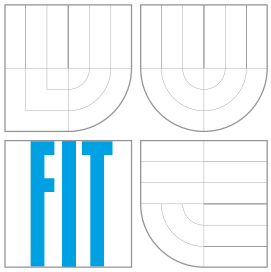
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

DETEKCE ÚTOKŮ DOS A DDOS POMOCÍ ZÁZNAMŮ NETFLOW

DETECTING DOS AND DDOS ATTACKS USING NETFLOW DATA

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

JAN HUŇKA

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. PETR MATOUŠEK, Ph.D.

BRNO 2013

Abstrakt

Tato bakalářská práce se zabývá využitím záznamů NetFlow pro detekci útoků DoS a DDoS. Na základě poznatků získaných analýzou útočného provozu je implementován plugin pro exportér sondy FlowMon, který sleduje různé heuristiky a podle nich stanoví míru podezření zdrojové IP adresy. Během testování je ověřeno, že plugin dokáže spolehlivě detekovat rozsáhlé útoky DoS a DDoS na živém provozu.

Abstract

This thesis deals with using NetFlow data for DoS and DDoS attacks detection. Based on the findings of the analysis of attack traffic a plugin for exporter of the FlowMon probe is implemented. It monitors several heuristics and based on them determines a level of suspicion of the source IP address. During testing, it was verified that the plugin is able to reliably detect large-scale DoS and DDoS attacks on live traffic.

Klíčová slova

DoS, DDoS, NetFlow, FlowMon, síťové útoky

Keywords

DoS, DDoS, NetFlow, FlowMon, network attacks

Citace

Jan Huňka: Detekce útoků DoS a DDoS pomocí záznamů NetFlow, bakalářská práce, Brno, FIT VUT v Brně, 2013

Detekce útoků DoS a DDoS pomocí záznamů NetFlow

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Petra Matouška, Ph.D.

.....
Jan Huňka
8. května 2013

Poděkování

Děkuji Ing. Petru Matouškovi, Ph.D. za vedení práce a cenné rady a Ing. Matěji Grégrovi za pomoc při testování. Dále také děkuji Ing. Petru Špringlovi a Mgr. Martinu Elichovi ze společnosti INVEA-TECH za spolupráci a rady při vývoji. V neposlední řadě samozřejmě děkuji rodině, přítelkyni a přátelům za veškerou podporu.

Obsah

Kapitola 1

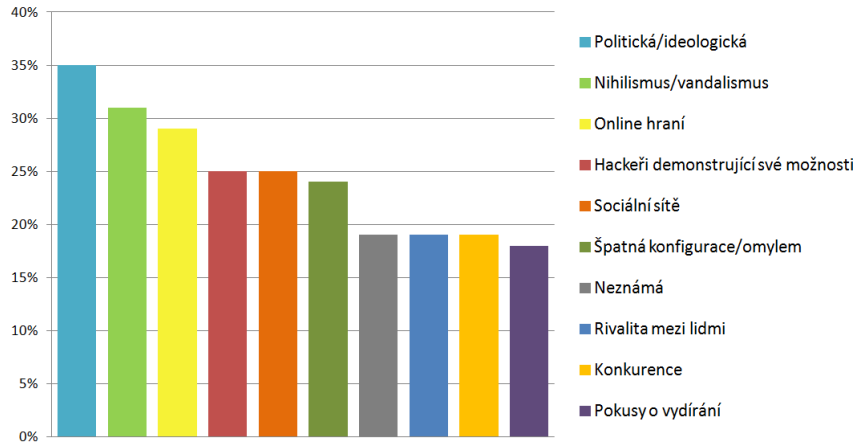
Úvod

Útoky na počítačové sítě se neustále vyvíjejí a jsou stále sofistikovanější. Velmi populární jsou v současnosti útoky typu Denial of Service (DoS) a to zejména díky své jednoduchosti a efektivnosti. Ovšem až s nástupem skupin hackerů jako Anonymous se povědomí o těchto útocích dostalo i mezi širší veřejnost. Popularita zejména distribuovaných variant DoS útoků se v posledních letech rapidně zvýšila. Mohou za to právě Anonymous a další skupiny, které je využívají jako prostředek k protestu a často tímto způsobem útočí na weby vlád a dalších organizací, s jejichž chováním nesouhlasí. Například v roce 2012 se staly cílem DDoS útoků vlády a organizace snažící se omezit svobodu internetu zákony jako SOPA, PIPA a ACTA [16]. Tomuto se souhrnně říká hacktivismus.

Útoky DoS a DDoS nejsou jen populárním nástrojem hacktivismu. Společnosti je mohou používat k poškozování své konkurence, či obyčejní hráči k pomstě proti svým rivalům. Motivů za použitím těchto útoků může být mnoho, ovšem společným faktorem zůstává, že se snaží znepřístupnit služby běžným uživatelům a jejich používání jen tak neskončí.

Ušetřeny nezůstaly ani české sítě. Dva nejznámější incidenty jsou z roku 2012 a 2013. V prvním případě proběhly útoky při prosazování mezinárodní dohody ACTA, která měla podle kritiků omezit práva občanů a svobodu internetu. Skupina Anonymous tehdy pomocí DDoS útoků zacílila na organizace ochrany autorských práv a vládní stránky, aby upozornila na kontroverznost této dohody a zabránila jejímu schválení. Druhý útok je z letošního roku a odehrával se v postupných vlnách celé tři dny. Cílem se za tu dobu staly zpravodajské weby, weby mobilních operátorů, bank a dopravních podniků. K útokům se nepřihlásila žádná skupina, ale zjistilo se, že byly prováděny ze zahraničí za použití počítačů infikovaných malwarem, které byly součástí velkého botnetu [18]. Nejčastější motivace útočníků jsou zobrazeny na obrázku 1.1. Zdrojem dat je výroční zpráva společnosti Arbor Networks [5], která se zabývá detekcí útoků DoS a DDoS. Samotný graf říká, kolik z účastníků průzkumu se setkalo v dané době s útokem, který byl takto motivován.

K útokům DoS a DDoS se mohou využívat počítače infikované malwarem, které bez uživatelova vědomí odesílají na pozadí velké množství dat směrem k cíli útoku. Dohromady tvoří obrovskou síť, kterou může útočník kdykoliv použít. Ještě větší hrozbu však představuje nárůst popularity útoků DoS a DDoS i mezi lidmi bez zvláštních technických znalostí. Díky hacktivismu se někteří lidé začali ztotožňovat s cíli skupin jako Anonymous a čím dál častěji se tak stává, že si sami dobrovolně instalují nebo skrz webové rozhraní využívají nástroje, které členové těchto skupin zveřejnili pro volné používání. Útoky DoS a DDoS už tedy nejsou jenom pro zkušené uživatele, ale může se do nich zapojit prakticky kdokoliv a to bez téměř jakékoliv technické znalosti této problematiky. Útoky tak mohou nabrat obrovských rozměru.



Obrázek 1.1: Motivace k útokům (D)DoS [5].

Ochrana proti útokům útokům DoS a DDoS je velmi náročná a útok musí být detekován co nejdříve, aby nestihl přetížit síťové prvky včetně těch provádějících detekci. Jednou z možností detekce probíhajících útoku je využití NetFlow záznamů. Ty jsou vytvářeny specializovanými sondami nebo směrovači podporujícími tuto technologii.

1.1 Cíle bakalářské práce

Tato práce vznikla ve spolupráci s brněnskou společností INVEA-TECH, která nabízí portfolio produktů FlowMon pro efektivní monitorování počítačových sítí na bázi síťových toků (NetFlow). Řešení FlowMon ovšem postrádá nástroj, který by zákazníky informoval o probíhajících útocích DoS a DDoS.

Cílem této práce je seznámit se s nejčastěji používanými typy útoků DoS a DDoS a jejich charakteristikami. Na základě získaných poznatků vytvořit plugin pro sondu FlowMon, který by na úrovni exportéru dokázal pomocí síťových toků detekovat nejpoužívanější typy těchto útoků a oznámil správci sítě IP adresy útočníků společně s dalšími užitečnými informacemi.

Při samotné implementaci pluginu by měl být kladen důraz zejména na to, aby použitá metoda detekce pokryla nejčastější typy útoků s minimálním výskytem falešných poplachů. Zároveň by měla být dostatečně rychlá pro nasazení pluginu na rozsáhlejší síti s velkým provozem. Plugin by se podle zadání měl dále zaměřit zejména na oblast distribuovaných DoS útoků, které jsou v současnosti nejpoužívanější a zároveň nejnebezpečnější.

1.2 Členění práce

Kapitola 2 se zabývá principy útoků DoS a DDoS a jejich nejčastějšími variantami. Kapitola 3 popisuje technologii NetFlow, její architekturu a použití. Zvažované metody detekce a návrh finální metody jsou popsány v kapitole 4. Seznámení se sondu FlowMon a popis implementace pluginu následuje v kapitole 5. Kapitola 6 analyzuje nejčastější typy útoků DoS a DDoS a je zde otestována úspěšnost a výkonnost pluginu při těchto útocích. Jako závěr a zhodnocení dosažených výsledků slouží kapitola 7.

Kapitola 2

Útoky DoS a DDoS

Pro úspěšnou detekci útoků DoS a DDoS je velmi důležité přesně definovat jejich typické vlastnosti. Pokud porozumíme tomu, jak se útoky projevují v různých charakteristikách síťového provozu, můžeme navrhnout metodu, která využije velkých změn v těchto charakteristikách k detekci samotného útoku.

Správce sítě může tyto útoky jednoduše identifikovat z grafu provozu na sítí (například na kolektoru NetFlow), kde jdou jednoznačně vidět místa s abnormálním vytížením sítě. Z toho vyplývá, že pro automatizaci tohoto procesu musíme najít způsob, jak identifikovat abnormálně velký provoz při útoku, stejně jako to zvládne správce sítě z obyčejného grafu.

Následující kapitola shrnuje typické vlastnosti útoků DoS a DDoS a jejich nejčastější druhy. Dále jsou zde popsány populární nástroje pro generování těchto útoků a možnosti obrany.

2.1 Základní charakteristiky útoků

Při útocích DoS a DDoS se útočníci snaží vyčerpat veškeré zdroje cíle útoku. Může jít o výpočetní výkon, přenosové pásmo nebo omezení datových struktur operačního systému. Cílem útočníků je narušit kritické služby organizace jako elektronické obchodování, e-mail nebo přístup k webovým stránkám, a znemožnit legitimním uživatelům přístup k těmto službám. O útocích DoS a DDoS tedy můžeme říct, že jde o útoky na dostupnost.

První útok DoS [10] se odehrál v roce 1996 proti ISP Panix v New Yorku. Šlo o SYN flood útok, který využíval podvrhnuté IP adresy a útočníci tak nemohli být vystopováni. Servery Panixu byly zahlceny průměrně 150 SYN pakety za sekundu a nemohly odpovídat na běžné požadavky. Jejich řešení tohoto problému spočívalo ve vytvoření speciální struktury, která udržovala napůl otevřená spojení, dokud neobdržela ACK paket. Spojení, která neobdržela ACK pakety, byla po vypršení zkráceného časového limitu zrušena.

Útoky se podle množství a typu útočníků dělí na:

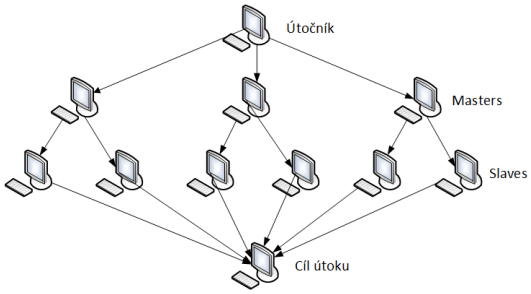
- **DoS**: Útok pocházející od jednoho útočníka.
- **DDoS (Distributed DoS)**: Útoku se účastní velké množství stanic, které koordinovaně zahlcují cíl útoku. Stanice se do útoků DDoS zapojují těmito způsoby:
  - **Infikováním**: Infikované počítače útočníka se skládají z master zombies a slave zombies. Útočník ovládá master zombies a ty spouštějí slave zombies. Proces probíhá tak, že útočník pošle příkaz k útoku pro master zombies a aktivuje tím

útočné procesy na těchto počítačích. Ty pošlou příkaz k útoku slave zombies, které zahájí útok DDoS proti cíli.

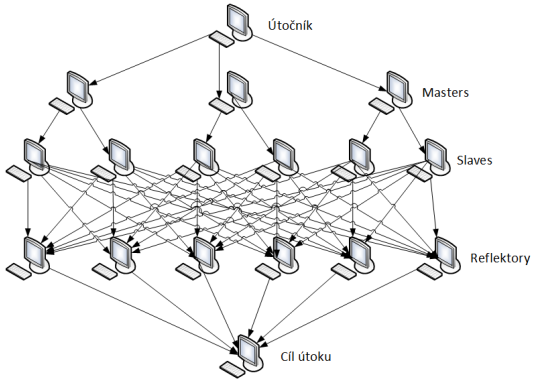
– **Dobrovolně**: Stále častěji se ale stanice zapojují do útoků dobrovolně. Jak jsem již zmínil v úvodu, útoky DDoS se stávají nástrojem hacktivismu. To je umožněno zejména snadnou dostupností nástrojů, které tyto útoky vytvářejí a jejich jednoduchostí. Do takového útoku se může zapojit opravdu každý.

Útoky DDoS mohou narůst do obrovských rozměrů, kdy je jakákoliv obrana velmi problematická. Často se také využívá podvrhování zdrojových adres k utajení identity útočníků.

- **DRDoS (Distributed Reflected DoS)**: Do určité chvíle se chovají stejně jako útoky DDoS s infikováním počítačů. Útočník má kontrolu nad master zombies, které ovládají slave zombies. Slave zombies na pokyn master zombies zasílají pakety s IP adresou cíle útoku jako zdrojovou adresou dalším neinfikovaným počítačům (reflektory) a vybízejí je tím připojit se k cíli útoku. Reflektory odpovídají cíli útoku, protože věří, že si vyžádal komunikaci [10]. Útok je tedy veden neinfikovanými počítači, které se účastní útoku bez vlastního vědomí.



Obrázek 2.1: Útok DDoS [10].

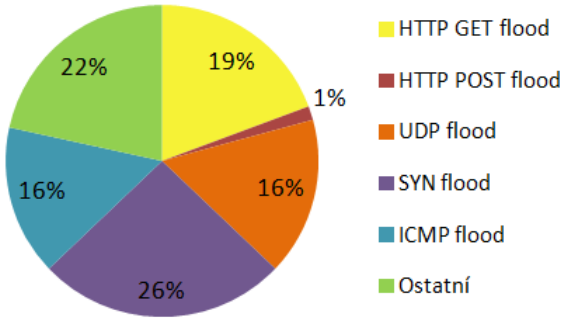


Obrázek 2.2: Útok DRDoS [10].

Některé formy útoků, zejména na aplikační vrstvě, se mohou tvářit jako naprosto legitimní provoz a je velmi obtížné rozlišit, které požadavky jsou součástí útoku, a které ne. Odpojování podezřelých uživatelů se může jednoduše minout účinkem, protože zablokujeme legitimní uživatele a útočníci částečně dosáhnou svého.

2.2 Nejčastější typy útoků

Existuje poměrně velké množství různých druhů útoků DoS a DDoS. V této práci jsem se zaměřil zejména na ty, které jsou v současnosti nejpoužívanější. Na obrázku 2.3 je uvedeno zastoupení nejčastějších typů útoků v prvním čtvrtletí roku 2013. Data grafu byla získána z pravidelné čtvrtletní zprávy společnosti Prolexic [13], která se zabývá obranou proti útokům DoS a DDoS. Mezi položku ostatní na grafu patří málo zastoupené útoky jako: ACK flood, PUSH flood, SSL GET flood, SSL POST flood a další.



Obrázek 2.3: Zastoupení jednotlivých typů útoků [13].

2.2.1 Útok SYN flood

Tento dnes již dobře známý útok [6] využívá základní vlastnosti protokolu TCP a to třífázové synchronizace (three-way handsake) [12]. Během normálního TCP spojení zahajuje komunikaci klient zasláním paketu TCP SYN na server. Pokud je server dostupný, tak odpoví paketem SYN/ACK. Klient poté odpovědí v podobě ACK paketu dokončuje třífázovou synchronizaci.

Při útoku SYN flood útočník posílá cílovému serveru velké množství SYN paketů, ale už neodpovídá paketem ACK. Server po určitou dobu čeká na potvrzovací ACK paket a během této doby má pro spojení alokovány zdroje. Tímto dochází ke vzniku velkého množství napůl otevřených TCP spojení. Pokud útočník zašle dostatečně velké množství SYN paketů, vyčerpá nakonec všechny dostupné zdroje serveru pro zahajování nových spojení a legitimní požadavky budou odmítány [14].

V případě, že útočník podvrhne zdrojovou adresu v IP hlavičce paketu, jsou odpovědi serveru SYN/ACK odesílány klientovi, který si komunikaci nevyžádal. Opravdový útočník se tím jednoduše zbaví nežádoucí zátěže své sítě a zároveň zůstane anonymní.

Při útoku SYN flood tedy vznikají jednopaketové toky s příznakem SYN v TCP hlavičce v poli TCP flags. Tyto útoky jsou velmi jednoduché a efektivní i v nedistribuované formě na menší servery. V případě velkých komerčních služeb jsou servery daleko výkonnější než stanice normálních uživatelů a je potřeba se uchýlit k DDoS útoku.

2.2.2 Útok UDP flood

Útok UDP flood využívá bezstavového transportního protokolu UDP k přenosu velkého množství paketů na náhodné nebo konkrétní porty cíle útoku. Ten se po přijetí takového paketu snaží zjistit, zda na tomto portu naslouchá nějaká aplikace. Pokud ne, odpoví paketem ICMP Destination Unreachable. Pro velké množství příchozích UDP paketů se tedy odesílá velké množství ICMP paketů. Útočník se tím snaží zahltit cíl útoku, který se stane nedostupným pro legitimní uživatele. Častým terčem útoků také bývají služby jako HTTP, DNS a VoIP. Zdrojová adresa útočníka může být podvrhnuta a opravdový vlastník této adresy obdrží všechny odpovědi serveru.

Útoky pomocí UDP protokolu jsou v současnosti často využívané a to i již zmiňovanou skupinou Anonymous. Velmi populárním nástrojem pro jejich generování je veřejně dostupný LOIC [19].

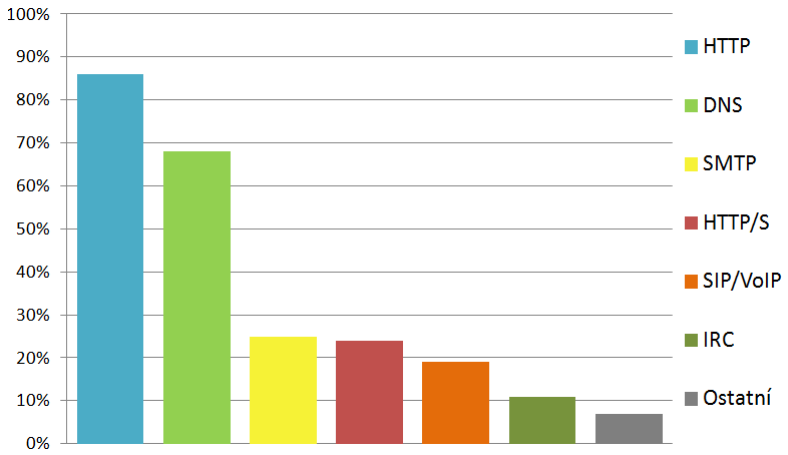
2.2.3 Útok ICMP flood

Útočník odesílá velké množství zpráv ICMP Echo Request co nejrychleji za sebou a snaží se zahltit cíl útoku. Využívá se toho, že každý takový požadavek vygeneruje okamžitou odpověď od serveru.

2.2.4 Útoky na aplikační vrstvě

Doposud zmíněné útoky pracovaly na čtvrté transportní vrstvě OSI modelu a byly založeny na využití omezeného přenosového pásma a výkonu. Proti některým z těchto útoků již dnes existují určité druhy obrany. V poslední době ale začaly být útoky DoS a DDoS sofistikovanější a postupně se přesouvají na sedmou aplikační vrstvu, kde mohou zaměřit konkrétní aplikační služby jako HTTP, DNS, VoIP a další. Zastoupení nejčastějších cílů útoků na aplikační vrstvě je zobrazeno na obrázku 2.4. Graf říká, kolik z účastníků průzkumu se setkalo za dané období s určitým útokem na aplikační vrstvě. Důvodů pro využití útoků na aplikační vrstvě může být několik:

- **Nenápadnější**: V porovnání s předchozími útoky jsou daleko nenápadnější a je obtížné je odlišit od běžného provozu.
- **Obchází jednu úroveň zabezpečení**: Ve většině případů bývají cílové aplikační služby dobře známé a jejich síťový provoz je ve firewallech povolen (např. HTTP nebo DNS provoz).
- **Peníze**: Pokud aplikace podporuje službu, která je hlavním zdrojem příjmů společnosti, stává se firma jednoduchým terčem vydírání a daleko ochotněji zaplatí za zastavení útoku, než aby přicházela o větší částky.



Obrázek 2.4: Nejčastější cíle útoků (D)DoS na aplikační vrstvě [5].

Mezi útoky na aplikační vrstvě patří například:

- **HTTP flood**: Jde o metodu, kdy se útočníci snaží cíl zahltit obrovským množstvím legitimních HTTP požadavků. Cílem je zablokovat konkrétní webovou stránku, aby nebyla přístupná běžným uživatelům.

- **Slowloris**: Tento typ útoků se stal populárním v roce 2009 díky vzniku volně dostupného nástroje Slowloris [15]. Ten se na rozdíl od předchozích útoků zaměřuje na pomalejší útoky DoS proti konkrétním službám. Tento koncept umožňuje jedinému počítači přetížit cizí web server za použití minimálního přenosového pásma a zároveň ponechat všechny ostatní služby serveru dostupné. Jde tedy o velmi nenápadný nástroj.

- **Slowpost**: Tato metoda je velmi podobná Slowlorisu s tím rozdílem, že se cílovému serveru zasílají kompletní HTTP POST požadavky a nekompletní data. V hlavičce HTTP požadavku útočník vyplní vysokou hodnotou pole Content-length, která udává, kolik dat má server při komunikaci očekávat. Útočník dále posílá data rychlostí například 1 byte každé dvě minuty, aby udržoval spojení neustále aktivní. Většina web serverů může přijmout až 2 GB v jediném HTTP POST požadavku a server by tedy čekal velmi dlouho na dokončení přenosu. Útočící aplikace mezitím však vytváří další taková spojení.

2.3 Nástroje útočníků

Spousta těchto nástrojů původně vznikla za účelem testování chování sítě pod velkou zátěží, ale po uvolnění začaly být využívány k přesně opačnému účelu. Napsat takový program není nic složitého, a proto jich existuje opravdu velké množství. Liší se zejména podporou různých typů útoků a možnostmi maskování identity útočníka. V průběhu zpracovávání této práce jsem se zaměřil na tyto nástroje:

- **LOIC (Low Orbit Ion Cannon)**: Jedná se o jeden z nejpoužívanějších a nejnebezpečnějších nástrojů pro útoky DoS a DDoS. Proslavil se zejména spojením se skupinou Anonymous, která ho využila k řadě veřejně známých útoků proti organizacím a společnostem jako FBI, Sony, PayPal, Mastercard, Visa a další [19].

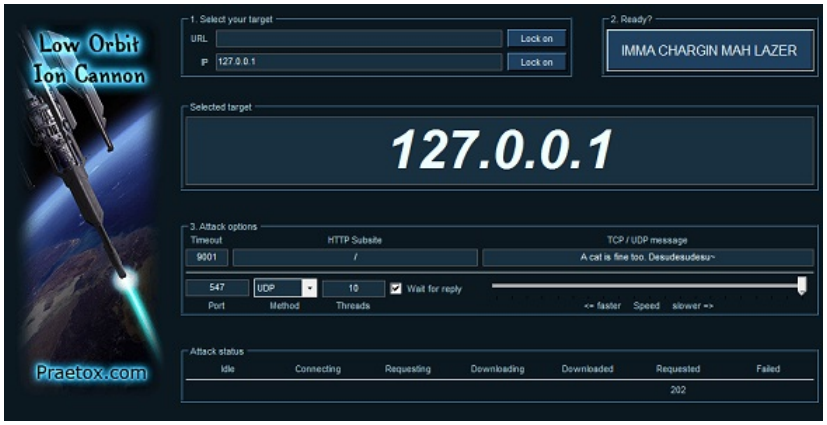
  Jde o velmi jednoduchý nástroj a právě to je jeho největší výhoda. V grafickém rozhraní je potřeba zadat pouze IP adresu nebo URL cíle a zvolit jednu z možností útoků TCP, UDP či HTTP. Mezi volitelná nastavení patří cílový port (defaultně 80), timeout, rychlost útoku, počet vláken či zpráva v transportním paketu. Rozhraní základní desktopové varianty nástroje LOIC je zobrazeno na obrázku 2.5. Existuje také webová verze tohoto nástroje, která je ještě jednodušší a je schopna provádět stejné útoky jako desktopová verze.

  Pokročilejší verze LOIC, která byla speciálně upravena skupinou Anonymous, umožňuje tzv. HIVEMIND mód [19]. Jde o možnost připojit klienta k IRC serveru, který tak může být ovládán vzdáleně. V tomto módu mohou tisíce počítačů na příkaz útočit na jedinou webovou stránku a vážně ji ohrozit.

  Jak jsem již zmínil, LOIC využívá tři různé typy útoků: TCP, UDP a HTTP. Všechny tři metody se chovají velmi podobně. Nejprve otevřou několik spojení k cílovému serveru a pak pravidelně za sebou odesílají požadavky, dokud server nepřetíží a ten přestane odpovídat.

  Nevýhodou LOIC jakožto útočného nástroje je absence možnosti maskování IP adresy útočníka podvrhnutím náhodnými adresami. Pokud není použit jiný nástroj pro podvrhnutí IP adresy, tak je při LOIC útoku v každém paketu obsažena opravdová adresa útočníka.

Kvůli popularitě a nebezpečnosti je právě LOIC nástrojem, na který jsem se při vývoji pluginu zaměřil. Výsledný plugin by měl zvládnout detekovat všechny typy útoků, které je LOIC schopen generovat.



Obrázek 2.5: Rozhraní desktopové varianty (D)DoS nástroje LOIC.

- **HPing**: Jednoduchá konzolová aplikace vycházející z nástroje ping, která má ale daleko více možností [17]. Může být využita ke generování velkého množství TCP, UDP a ICMP provozu. Umožňuje nastavovat i další vlastnosti paketů jako TCP flags. Velkou výhodou je, že dokáže maskovat zdroj útoku pomocí podvrhnutí zdrojové IP adresy. Tento nástroj budu využívat při analýze a testování pro generování ICMP a SYN flood útoků. Existuje samozřejmě více podobných nástrojů, ale chování při těchto dvou útocích bývá prakticky identické u všech nástrojů, a proto jsem HPing vybral jako jejich zástupce.

- **Slowloris**: Zaměřuje se na pomalé útoky DoS. Udržuje si otevřená spojení odesíláním částečných HTTP požadavků přes úplné TCP spojení. Posílá další hlavičky v pravidelných intervalech, aby zabránil uzavření socketů. Tomuto jsou náchylné zejména servery využívající vlákna, protože mívají omezený počet vláken, aby zabránily vyčerpání paměti.

  Nevýhodou tohoto nástroje je, že musí čekat, až budou dostupné všechny sockety, než je může zabrat pro sebe. Pokud jde o vytíženou stránku, může takový útok chvíli trvat. Když některý uživatel stihne navázat spojení dříve než Slowloris, může stále získat přístup ke stránce pod útokem. V počáteční fázi je to tedy závod o čas, který ale Slowloris poměrně rychle vyhraje. Naopak jeho výhodou je, že vyžaduje pouze několik stovek požadavků v delších a pravidelných intervalech a po útoku se web server může vrátit do původního stavu prakticky okamžitě.

  Tento nástroj není zahrnut ve fázi testování a není uvažován ani při výběru heuristik. Je o velmi specifický typ útoku a je tedy potřeba stanovit novou množinu heuristik, které by s tímto útokem počítaly. To se nabízí jako případné rozšíření a stejně tak i detekce jiných typů útoků.

2.4 Detekce a ochrana proti útokům

Existuje mnoho různých přístupů k detekci útoků DoS a DDoS, ale všechny vycházejí z těchto základních strategií:

- **Detekce pomocí vzorů (Signature-based detection)**: Tyto metody porovnávají zachycený provoz na síti se vzory známých útoků, které jsou uložené v databázi. Mohou jednoduše a spolehlivě detekovat známé útoky, ale nerozpoznají nezdokumentované útoky a databáze musí být neustále aktualizována.
- **Detekce anomálií (Anomaly-based detection)**: Metody založené na detekci anomálií srovnávají parametry pozorovaného provozu na síti s obrazem normálního provozu. Musí se udržovat aktuální model normálního provozu a zvolit prah pro odlišení anomálie od běžného provozu.
- **Hybridní systémy**: Kombinace předchozích strategií.

Po úspěšné detekci útoku na něj reagují různé reaktivní obranné systémy, které se snaží omezit následky. Samotná obrana proti útokům DoS a DDoS je velmi složitá a účinná pouze částečně. Využívají se například tyto metody [10]:

- **Firewall a ACL (Access Control List)** může definovat jednoduchá pravidla pro filtrování provozu.
- **IPS (Intrusion Prevention System)** jsou zařízení, která procházejí v reálném čase síťový provoz. Slouží k detekci a zachycení podezřelého provozu (např. Cisco IPS).
- **Blackhole routing** směruje provoz na rozhraní null, kde je zahozen. Není ovšem vždy možné odloučit útočný provoz od běžného provozu.
- **Sinkhole routing** směruje podezřelý provoz na jiný router, který je dimenzován, aby útoku odolal. Zde se dále analyzuje a útočný provoz je filtrován. Minimalizuje se tak riziko, zatímco se provádí prošetření dat.
- **Honeypot** je celá síť, která je vytvořena k tomu, aby přilákala útočníky. Uvnitř této sítě je zaznamenávána veškerá aktivita a zachytává útočníky.
- **Ingress filtering** filtruje podvrhnuté IP adresy, ale nic nezmůže proti validním zdrojovým IP adresám.
- **Egress filtering** monitoruje odchozí provoz na útočná data.
- **Load Balancers** jsou specializovaná zařízení, která ubírají zátěž serveru a mohou zpracovávat velké množství dat bez ztráty výkonu. Mohou implementovat například SYN cookies [6], které jsou účinnou obranou proti útokům typu SYN Flood.
- **Distribuovanost** znamená, že systémy jsou rozděleny na více míst a útočník by tak musel napadnout všechny najednou.

Kapitola 3

NetFlow

S obrovským nárůstem využívání IP síti se objevila nutnost nové technologie, která by efektivně zaznamenávala využívání síťových a aplikačních zdrojů. To umožnil až vznik technologie NetFlow, která se od té doby stala celosvětovým standardem v oblasti analýzy a sledování síťového provozu.

Při práci se sondou FlowMon, která je postavena na technologii NetFlow, je potřeba detailně porozumět architektuře a informacím, které je možné z NetFlow získat. Jde o velmi mocný nástroj, který správnou analýzou síťových toků může zachytit jakékoliv podezřelé chování síťového provozu. Právě architekturou a vlastnostmi technologie NetFlow se bude zabývat následující kapitola.

3.1 Využití NetFlow

NetFlow je proprietární standard vytvořený firmou Cisco (specifikace v RFC 3954 [4]), který umožňuje správcům sítí detailní pohled na chování síťového provozu na základě sledování tzv. toků. V tocích se neukládají informace o přenášených datech, což značně snižuje zátěž na paměť a výkon. Díky tomu je NetFlow vhodný zejména pro vysokorychlostní sítě. Informace získané z NetFlow záznamů se dají použít pro [2]:

- **Monitorování sítí**: NetFlow data umožňují monitorování sítě prakticky v reálném čase a zároveň sledují dlouhodobé statistiky provozu. Můžeme je využít k vizualizaci trendů provozu na jednotlivých síťových zařízeních nebo na celé síti, což umožňuje detekci a rychlé řešení různých problémů.
- **Zachycení útoků v reálném čase**: Na úrovni exportéru se dají rychle detekovat například útoky DoS a DDoS, červi, viry a skenování. Změny v chování sítě způsobují anomálie, které bývají v NetFlow datech zřetelné. Na úrovni kolektoru můžeme také zpětně dohledávat různé incidenty.
- **Monitorování aplikací**: Díky NetFlow získáme detailní přehled o aplikacích používaných na síti. Tuto informaci lze použít k plánování nových služeb, alokování zdrojů pro aplikační služby jako web server a VoIP. Zároveň můžeme takto monitorovat aktivity uživatelů na síti, zjišťovat, kdo síť nejvíce vytěžuje, a zda nejsou aplikace, které uživatelé používají, nežádoucí (např. P2P klienti).
- **Účtování**: Jednoduše můžeme NetFlow data využít k fakturování služeb podle počtu přenesených dat, využité šířky pásma, využité služby nebo denní doby. Lze takto

kontrolovat i dodržování podmínek uvedených v SLA (Service Level Agreement).

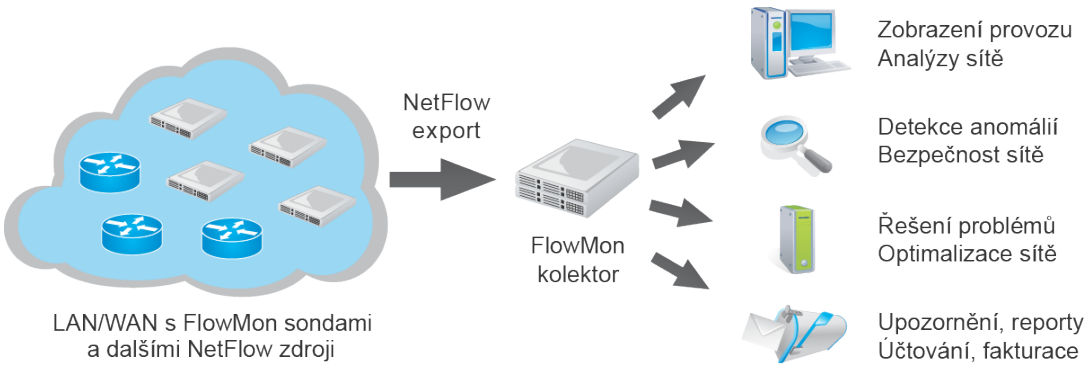
- **Rozvoj sítě**: Zachycené statistiky mohou být využity k plánování rozvoje sítě a upgradování či přidání nových síťových zařízení. Minimalizují se tím náklady a maximalizuje se výkon a spolehlivost sítě. Umožňuje také ověřovat úroveň kvality služeb (QoS).

Existuje několik verzí NetFlow, z nichž nejpoužívanější jsou verze 5 a 9. Verze 9 se liší zejména přidáním L2 informací a tím, že struktura záznamů je dána šablonou. Nejnovější verzí je Internet Protocol Flow Information eXport (IPFIX) [3], který vychází verze 9.

3.2 Architektura NetFlow

Základní prvky architektury systému NetFlow jsou [4]:

- **Bod pozorování** je místo v síti, kde mohou byt pozorovány IP pakety, například rozhraní směrovače.
- **Tok** je definován jako jednosměrná posloupnost paketů mající společnou vlastnost a procházející bodem pozorování za určitý čas. Pakety jednoho toku se shodují minimálně ve zdrojové a cílové IP adrese, zdrojovém a cílovém portu, názvu rozhraní, typu L3 protokolu a položce Type of Service.
- **NetFlow záznam** ukládá informace o toku, který je zachycen v bodě pozorování.
- **Exportér** je zařízení (směrovač, specializovaná HW nebo SW sonda), které agreguje procházející pakety a vytváří z nich toky. Ty jsou poté exportovány v podobě NetFlow záznamů protokolem NetFlow na kolektor.
- **Protokol NetFlow** je komunikační protokol pro odesílání dat mezi exportérem a kolektorem.
- **Kolektor** přijímá data z jednoho nebo více exportérů a ukládá je na disk. Před uložením mohou být záznamy agregovány. Nad uloženými daty se dá provádět grafická reprezentace, zjišťovat různé statistiky či provádět dotazy.

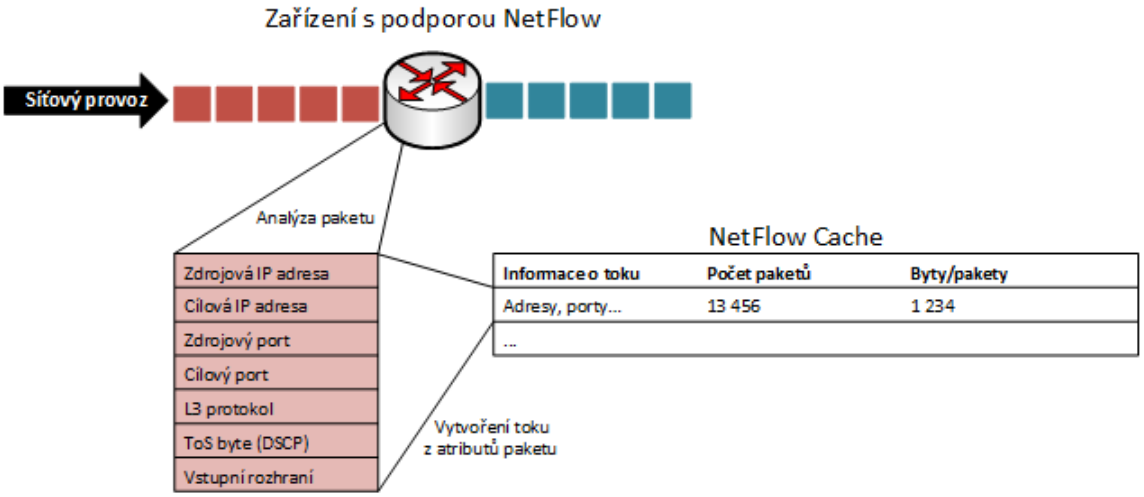


Obrázek 3.1: NetFlow architektura se sondami FlowMon [8].

3.3 Princip činnosti NetFlow

Na úrovni exportéru je vytvořena dynamická paměť NetFlow cache, která uchovává záznamy o aktivních tocích. Nový záznam je vytvořen na základě přijetí prvního paketu toku a modifikuje se na základě přijetí dalšího paketu, který patří do tohoto toku. Tento proces je zobrazen na obrázku 3.2. Záznamy v určitém okamžiku expirují a jsou periodicky odesílány na kolektor. NetFlow protokol je zároveň velmi úsporný a tvoří pouze přibližně 1.5 % provozu na směrovači. Expirace záznamů probíhá na základě následujících pravidel:

- **Vypršení neaktivního časovače**: Když je tok po specifikovanou dobu neaktivní (FlowMon defaultně 10 sekund).
- **Vypršení aktivního časovače**: Tok překročí maximální dobu existence (FlowMon defaultně 180 sekund).
- **Detekce konce toku**: U TCP příznaky RST nebo FIN.
- **Zaplnění NetFlow cache**



Obrázek 3.2: Princip činnosti NetFlow sondy [1].

Exportované záznamy jsou spojeny do exportovacích paketů, které se ve verzi NetFlow 5 nebo 9 mohou skládat až ze třiceti záznamů [2] a jsou odeslány protokolem UDP na kolektor. Kvůli základní vlastnosti protokolu UDP, tj. bezstavovosti, může při přenosu dojít ke ztrátě dat. NetFlow verze 9 je navržen tak, aby byl nezávislý na transportním protokolu.

V tabulce 3.1 jsou uvedena pole hodnot, která jsou k dispozici v záznamech NetFlow a lze je použít k analýze každého toku. Konkrétně jde formát NetFlow v5 záznamu.

Byty	Obsah	Popis
0-3	srcaddr	Zdrojová IP adresa
4-7	dstaddr	Cílová IP adresa
8-11	nexthop	IP adresa next hop routeru
12-13	input	Index vstupního rozhraní
14-15	output	Index výstupního rozhraní
16-19	dPkts	Počet paketů v toku
20-23	dOctets	Počet bytů v toku
24-27	First	Čas počátku toku
28-31	Last	Čas přijmutí posledního paketu toku
32-33	srcport	Zdrojový TCP/UDP aplikační port
34-35	dstport	Cílový TCP/UDP aplikační port
36	pad1	Nevyužité byty
37	tcp_flags	TCP příznaky (SYN, FIN, ACK, RST)
38	prot	IP protokol (TCP, UDP)
39	tos	Type of Service (ToS)
40-41	src_as	Zdrojový autonomní systém
42-43	dst_as	Cílový autonomní systém
44	src_mask	Maska zdrojové podsítě
45	dst_mask	Maska cílové podsítě
46-47	pad2	Nevyužité byty

Tabulka 3.1: NetFlow záznam verze 5 [2].

Kapitola 4

Návrh systému pro detekci útoků pomocí NetFlow

Tato práce byla již od začátku koncipována zejména jako výzkumná práce. Cílem bylo navrhnout metodu detekce, která by fungovala v reálném čase a v omezených podmínkách exportéru. Navrhnout ale takovou metodu, která by nebyla příliš výpočetně náročná a zároveň by dokázala útoky detekovat s omezenými informacemi, se ukázalo být poměrně obtížné.

Během seznamování se sondou FlowMon jsem dále narazil na problém spojený s pamětí NetFlow cache. Zatím zde totiž neexistuje programové rozhraní pro přímou práci s pamětí NetFlow cache. I zadání této práce spoléhá na procházení této paměti, takže bylo nutné zaměřit se na metody, které by nepotřebovaly procházet celou paměť nebo si uchovávat vlastní a do jisté míry redundantní data v jiné datové struktuře.

4.1 Poměry odchozího a příchozího provozu

První metoda, kterou jsem zvažoval, využívá poměrů mezi příchozím a odchozím počtem paketů jednotlivých transportních protokolů [14]. Množství příchozího a odchozího provozu se za normálních podmínek mění současně. Když však dojde k útoku do sledované sítě, začne převažovat příchozí provoz a poměr se změní. Jde o nenáročnou a poměrně efektivní metodu, která ovšem plně nevyužívá výhody, které NetFlow poskytuje. Pro detekci mohou být použity poměry mezi odchozími a příchozími pakety protokolů TCP, UDP, ICMP nebo všech protokolů společně a to podle rovnice 4.1.

$$R = \frac{\sum_{i=1}^{n} odchozi\,pakety\,protokolu}{\sum_{i=1}^{n} prichozi\,pakety\,protokolu}$$

Rovnice 4.1: Vzorec poměru odchozího a příchozího provozu.

Pro ověření této metody jsem spustil na školním serveru jednoduchý skript, který po dobu jednoho týdne sbíral data o odchozím a příchozím provozu protokolem SNMP a ukládal je do souboru. Po vyhodnocení dat jsem dospěl k závěru, že tato metoda by poměrně obtížně rozlišovala útoky od náhle zvýšeného provozu, což se v naměřených datech projevilo při pravidelných nočních zálohách. Ukázka naměřených dat je uvedena v příloze D.

Další problémem této metody je určení směru daného toku. Protokol SNMP sice sám sbírá informace o příchozím a odchozím provozu, ale protokol NetFlow směr toku už ze své definice neuvažuje. Existují dvě možnosti, jak rozlišit směr toku u NetFlow:

- Projít celou paměť NetFlow cache a ke každému toku dané zdrojové IP adresy najít všechny toky, které mají tuto IP adresu jako cílovou. Vzhledem k tomu, že sonda FlowMon neumožňuje přístup do NetFlow cache, je tento přístup nerealizovatelný.
- Pokud má sonda dva monitorovací porty, lze každý směr přivést na odlišná rozhraní a identifikovat je podle rozhraní uvedeném v NetFlow záznamu. Dva monitorovací porty ovšem nemají všechny FlowMon sondy a bylo by nutné spoléhat na určitou topologii celého zapojení sondy do sítě, což by velmi ztížilo testování a praktické nasazení.

Po zvážení těchto faktů jsem se rozhodl zaměřit na obecnější statistické metody detekce, které by provoz vnímaly jako celek.

4.2 Korelace mezi pakety, toky a byty

Diplomová práce Matěje Plcha z ČVUT [11] se zabývá závislostmi mezi počtem přenesených paketů, toků a bytů a na základě nich provádí detekci útoků DoS a DDoS na úrovni kolektoru. Je zde dokázáno, že v běžném provozu existuje mezi těmito metrikami přímá závislost. Při útoku je ale porušena, což může být použito jako zajímavá a obecná metoda detekce různých typů útoků.

Pokusil jsem se ověřit možnost využití korelace mezi pakety, byty a toky pro detekci útoků DoS a DDoS na úrovni exportéru. Ten má oproti kolektoru k dispozici pouze aktuální provoz a je zde potřeba dbát na co nejmenší zpoždění detekce útoků, což se záhy ukázalo být největším problémem tohoto přístupu.

K vyjádření korelace mezi metrikami x a y je využit Pearsonův korelační koeficient [7]. Proměnné x a y mohou nezávisle na pořadí reprezentovat jakékoliv dvě metriky z trojice pakety, toky, byty. Výpočet koeficientu pro sérii naměřených hodnot 1..n je definován následovně:

$$r = \frac{\sum_{i=1}^{n}[(x_i - \overline{x})(y_i - \overline{y})]}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2 \sum_{i=1}^{n}(y_i - \overline{y})^2}}$$

Rovnice 4.2: Vzorec Pearsonova korelačního koeficientu.

Hodnoty $\overline{x}$ a $\overline{y}$ značí střední hodnoty x, y v naměřeném vzorku. Korelační koeficient nabývá hodnotu z intervalu <-1;1>, která určuje míru závislosti mezi metrikami. Při běžném provozu se jeho hodnota blíží k jedné, což značí přímou závislost. Během útoku se pak snižuje k nule. Pomocí testování je potřeba stanovit prahovou hodnotu korelačního koeficientu, která bude při překročení značit útok.

Tuto metodu jsem testoval na reálném provozu sítě Listovy koleje, abych odhalil, zda korelační koeficient potvrdí lineární závislost. Během testu byly zaznamenávány počty paketů, bytů a toků za 1 sekundu a to vždy po dobu jedné minuty, což poskytuje 60 hodnot každé metriky pro výpočet v každém časovém okně. Ukázalo se ale, že tato doba není dostačující. Výsledné koeficienty jsou uvedeny v tabulce 4.1. Jde vidět, že i bez útoku se koeficienty

blížily nule, protože i velmi malá změna v provozu dokázala výrazně ovlivnit výsledný korelační koeficient. To je způsobeno nedostatkem naměřených hodnot. Doba měření by se samozřejmě dala prodloužit, ale ani okno trvající 3 minuty se neukázalo být dostačující a detekce by v tomto případě trvala příliš dlouho a to není na úrovni exportéru žádoucí. Po testování jsem došel k závěru, že tuto metodu by bylo vhodnější provozovat na úrovni kolektoru pro zpětné zkoumání provozu.

pakety/toky	**byty/toky**	**byty/pakety**
0.396482	0.236539	0.980533
0.015428	0.215139	0.964550
0.325013	0.161508	0.978241
0.179039	0.067198	0.953046
0.462534	0.279318	0.973301

Tabulka 4.1: Naměřené korelační koeficienty za 5 minut.

4.3 Bodový systém hodnocení heuristik

Zkušenosti získané při testování metody sledování korelace mi přinesly několik důležitých poznatků. Ověřil jsem si, že exportér není příliš vhodné místo pro statistické metody detekce kvůli omezení dostupných informací v danou chvíli a také, že úspěšná detekce by neměla spoléhat pouze na jeden ukazatel. Proto jsem se rozhodl poslední metodu postavit na sledování různých heuristik, které jsou ohodnoceny bodovým systémem. Spoléhání na několik heuristik najednou totiž razantně zvyšuje přesnost detekce. Stavový diagram celého systému je zobrazen na obrázku 4.1.

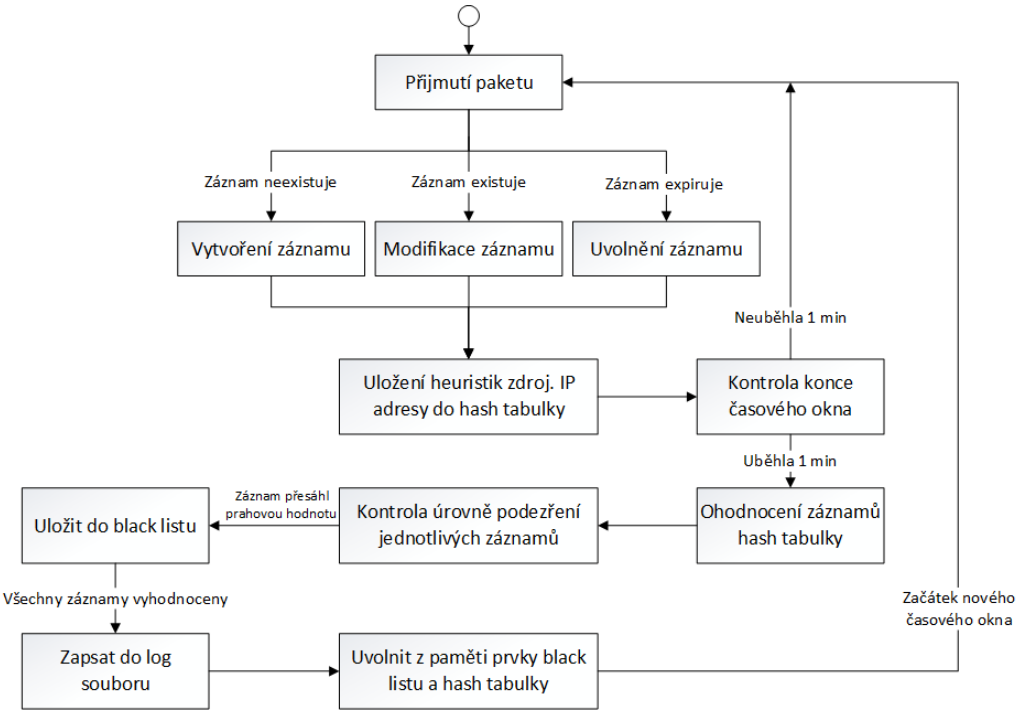
Vzhledem k nedostatku dostupných informací jsem si vytvořil vlastní paměť typu hash tabulka, která by udržovala po určitou dobu sbírané informace o zdrojových IP adresách. Abych jen nekopíroval data, která jsou již uložená v NetFlow cache, ukládám pouze informace, které budou sloužit jako heuristiky při vyhodnocení podezření dané zdrojové IP adresy. Záznam v hash tabulce je adresován zdrojovou IP adresou a všechna ukládaná data se sčítají pro všechny toky z této adresy po dobu jednoho časového okna. Na konci tohoto okna je tedy k dispozici kumulovaná informace o tom, jaký provoz IP adresa po tuto dobu generovala.

Po uplynutí časového okna se projde celá hash tabulka a záznamy jednotlivých zdrojových IP adres jsou podrobeny hodnotící funkci. Ta všechny heuristiky porovná s jejich prahovými hodnotami a přidělí záznamu úroveň podezření. Pokud je tato úroveň větší nebo rovna dvěma bodům, je záznam IP adresy uložen do lineárního seznamu black_list. Po vyhodnocení celé hash tabulky je black_list vypsán do logovacího souboru, všechny záznamy hash tabulky i black_listu jsou uvolněny z paměti a celý proces začíná od začátku.

Jak jsem uvedl v kapitole 2 na obrázku 2.3, jedním z nejpoužívanějších typů útoků jsou v současnosti útoky typu HTTP flood, a proto jsem se při detekci zaměřil zejména na útoky na port 80. Útoky na aplikační vrstvě mohou být vedeny i proti jiným službám jako DNS, pro které může být velmi jednoduše použito stejné řešení jako u HTTP útoků.

Útočné IP adresy jsou identifikovány na základě následujících heuristik:

- **Počet paketů**: Pokud je počet odeslaných paketů IP adresy menší než prahová hodnota, pak se tento záznam při hodnocení neuvažuje a jeho bodové ohodnocení je 0.



Obrázek 4.1: Stavový diagram systému hodnocení heuristik.

Takto malé přenosy nejsou nebezpečné pro infrastrukturu a je zbytečné s nimi dále pracovat.

- **Poměr TCP pakety/pakety**: Tato heuristika určuje poměr zastoupení TCP paketů z celkového počtu všech paketů, které sonda přijala v daném časovém okně [14]. Pokud TCP provoz přesáhne určitou hranici, dochází pravděpodobně k útoku a všem záznamům v hash tabulce se zvýší úroveň podezření o 1. To samé platí i pro protokoly UDP a ICMP.
- **Poměr UDP pakety/pakety**: Poměr zastoupení UDP paketů z celkového počtu všech paketů, které sonda přijala v daném časovém okně.
- **Poměr ICMP pakety/pakety**: Poměr zastoupení ICMP paketů z celkového počtu paketů všech paketů, které sonda přijala v daném časovém okně.
- **Poměr toky/pakety (port 80)**: Poměr toků a paketů odeslaných IP adresou na port 80. Při HTTP flood útocích se objevuje velké množství paketů na tok. Pokud je tento poměr menší než prahová hodnota, zvýší se podezření IP adresy.
- **Počet UDP toků a ICMP Destination Unreachable paketů**: Pokud IP adresa vygeneruje za časové okno příliš mnoho UDP toků, může jít o UDP flood na náhodná čísla portů. To se projeví také velkým počtem ICMP Destination Unreachable paketů v opačném směru.
- **Počet jednopaketových SYN toků**: Při útoku SYN flood vzniká velké množství jednopaketových TCP toků s příznakem SYN. Bodové ohodnocení této heuristiky je 2, protože se ukázalo, že jde o velmi spolehlivý způsob detekce, který nevyžaduje další zpřesnění.

- **Počet Echo Request paketů**: ICMP flood útoky se vyznačují velkým množstvím ICMP Echo Request paketů a tato heuristika sleduje jejich počet za časové okno.

Prahové hodnoty heuristik musí být stanoveny na základě testování chování heuristik při provozu bez útoku a s útokem. Výhodou tohoto systému je jednoduchost a snadná modifikovatelnost. Pro zahrnutí nového typu útoku stačí pouze přidat do struktury záznamu hash tabulky novou proměnnou, ukládat do ní požadovanou heuristiku a nastavit její prahovou hodnotu v hodnotící funkci. Plugin už sám automaticky vyhledá a ohodnotí všechny záznamy, které prahovou hodnotu přesahují. Lze tedy velmi snadno rozšířit množinu detekovaných útoků nebo zpřesnit stávající detekci.

Kapitola 5

Implementace pluginu pro sondu FlowMon

NetFlow sonda je alternativou ke sběru toků na směrovačích a přepínačích s podporou technologie NetFlow. Jde o pasivní monitorovací zařízení, které pozoruje protékající síťový provoz a na základě něj vytváří toky, které jsou po expiraci odesílány do externího kolektoru. Tento přístup má oproti zachytávání toků na směrovačích několik výhod. Tou nejdůležitější je, že odpadává nutnost vzorkování paketů na směrovači z důvodu ušetření zátěže a ten se může věnovat pouze vlastní činnosti.

Sonda se zapojí do monitorované linky a dostane se k ní kopie procházejícího provozu, který dále zpracovává. Tyto sondy jsou transparentní na L2 i L3 vrstvě [9] a těžko se tak stávají cílem útoku. Zapojují se nejčastěji na vstupní a výstupní body sítě a nejvytíženější a kritická místa. Jde tedy o ideální místo pro detekci útoků jako DoS a DDoS. V této kapitole budu popisovat vlastnosti a vývojové prostředí sondy FlowMon a také princip fungování celého pluginu.

5.1 Popis a architektura sondy FlowMon

FlowMon sonda je součástí portfolia produktů FlowMon společnosti INVEA-TECH, které tvoří kompletní řešení pro monitorování sítí pomocí toků, skládající se z FlowMon sondy, FlowMon kolektoru a FlowMon pluginů. Základní vlastnosti FlowMon sondy jsou [8]:

- Výkonná autonomní NetFlow sonda.
- Standardní a hardwarově akcelerované modely.
- Zpracování dle rychlosti linky bez ztráty paketů (není potřeba vzorkování).
- Podpora pro 10/100/1000 a desetigigabitový Ethernet.
- Schopnost zpracovat až 512 000 souběžných toků a 6 milionů paketů za sekundu.
- 1x 10 Gb/s nebo až 4x 10/100/1000 monitorovací rozhraní.
- Jednoduchá konfigurace webovým rozhraním.
- Kompatibilní s nejrozšířenějšími NetFlow kolektory jiných výrobců.
- Podpora pro IPv4, IPv6, VLAN and MPLS.

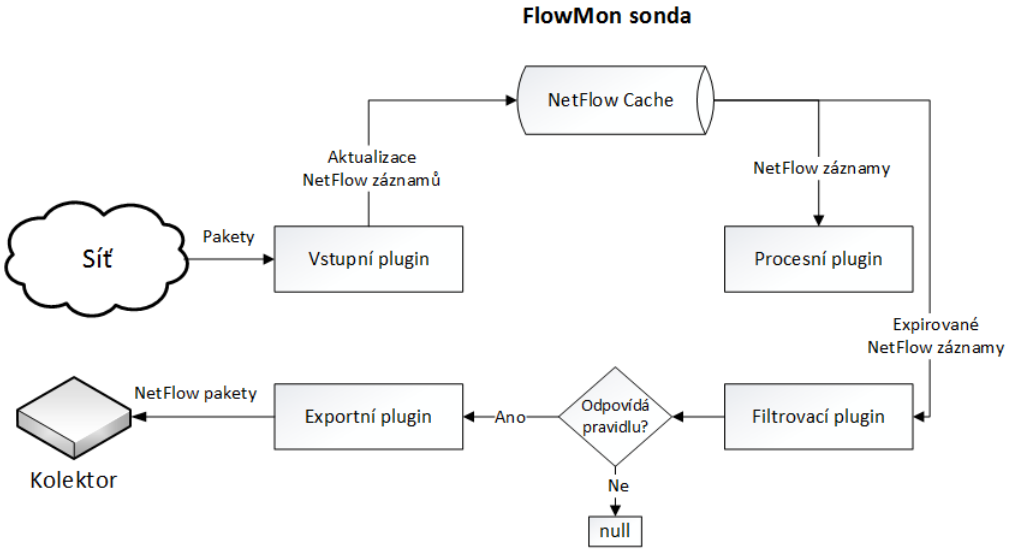
- Export dat ve formátech NetFlow v5, v9, IPFIX

Exportér sondy FlowMon je program implementovaný v jazyce C, který je zodpovědný za zachytávání paketů, jejich parsování, tvorbu NetFlow záznamů, jejich ukládání do paměti NetFlow cache a následný export záznamů na kolektor.

NetFlow data exportovaná sondou jsou dále zpracována FlowMon kolektorem, který je dlouhodobě ukládá a uživatelé je mohou dále analyzovat a vizualizovat pomocí FlowMon monitorovacího centra. FlowMon sondy také obsahují vestavěný kolektor pro seznámení s technologií NetFlow nebo k použití v menších až středních sítích.

5.2 Tvorba pluginů pro FlowMon exportér

Sonda FlowMon je postavena nad operačním systémem Linux a to konkrétně nad distribucí CentOS [9]. Do exportéru lze tvořit pluginy rozšiřující funkcionalitu a ovlivňovat tak jeho činnost. Ty poskytují pokročilou analýzu NetFlow dat a další zajímavé informace o síťovém provozu. Tvorbu těchto pluginů podporuje firma INVEA-TECH svým komunitním programem, kde jsou dostupné informace pro programátory, kteří se vývojem pluginů zabývají. Vývoj takových pluginů probíhá ve virtuální sondě FlowMon, která samozřejmě nemá takový výkon jako hardwarová sonda, ale pro vývoj těchto pluginů dostačuje.



Obrázek 5.1: Architektura pluginů pro FlowMon exportér.

Tvorba pluginů probíhá podle přesně stanoveného schématu. Je potřeba vytvořit tyto čtyři základní pluginy, které společně tvoří požadovanou funkcionalitu:

- **Vstupní plugin (input)**: Zachytává pakety ze síťového rozhraní nebo ze souborů PCAP. Z těchto zachycených dat se vytváří NetFlow záznamy v paměti NetFlow cache pomocí interní funkce nebo je může plnit sám programátor. Ke každému NetFlow záznamu se mohou přidat i další volitelná data, která jsou poté přístupná společně se základními daty tohoto toku.

  Příchozí paket je přidán k příslušnému záznamu v paměti NetFlow cache vypočtením hodnoty hash z vlastností, které identifikují síťový tok. Tato hodnota identifikuje

záznam, ke kterému paket patří. Velikost NetFlow cache je ve výchozím stavu 2^{19} záznamů (524 288 záznamů).

- **Procesní plugin (process)**: Umožňuje pracovat s jednotlivými záznamy toků ve chvíli při jejich vytvoření (tj. při příchodu prvního paketu), při příchodu dalšího paketu, který patří k tomuto toku a ihned po expiraci záznamu. V tomto pluginu lze detekovat v reálném čase různé útoky a získat informace vždy o jednom toku v danou chvíli.
- **Filtrovací plugin (filter)**: Definuje filtry pro expirované toky, které se chystají k exportu. Na základě tohoto filtru může být tok buď propuštěn k exportu nebo zamítnut. Implementace tohoto pluginu není povinná.
- **Exportní plugin (export)**: Zde se vytvářejí pakety s NetFlow záznamy a ty se odesílají na kolektor v požadovaném formátu (nejčastěji v5, v9 a IPFIX pakety). Záznamy se také mohou tisknout pouze na standardní výstup.

Každý z těchto pluginů má vlastní hlavičkový soubor, kde jsou deklarované jednotlivé základní funkce, které se v pluginech používají. Je pak na tvůrci pluginu, aby dodal definici těchto funkcí a tím vytvořil svůj vlastní plugin. Na pomoc je k dispozici několik interních funkcí exportéru (např. funkce na parsování paketů, která vrací jejich hash pro NetFlow cache).

5.3 Popis implementace pluginu

Detekce útoků DoS a DDoS se v tomto případě odehrává výhradně na úrovni procesního pluginu. Vstupní plugin pouze načítá pakety z monitorovacího rozhraní sondy a zpracovává je pomocí interní funkce `record_process_packet`, která vrátí hash hodnotu a podle ní je aktualizován či vytvořen příslušný záznam v paměti NetFlow cache. Filtrovací plugin není v tomto případě potřebný a exportní plugin může v případě potřeby vytisknout záznamy na standardní výstup. O exportování záznamů na kolektor se stará sonda.

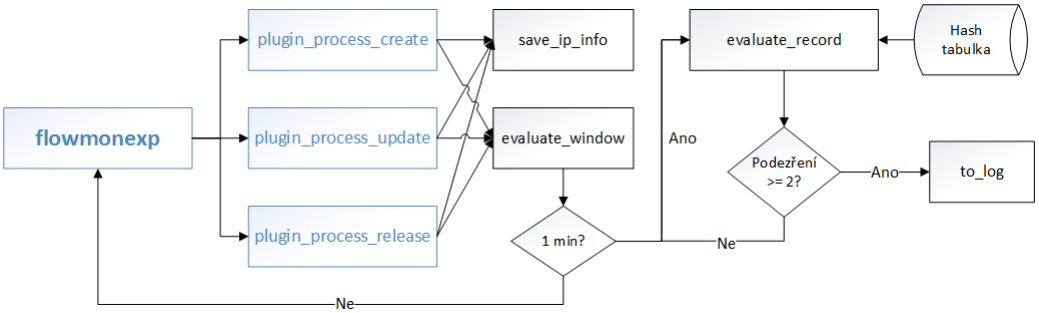
```
typedef struct hash_table_record
{
        uint32_t *ipv4_addr;
        uint32_t flows;
        uint32_t packets;
        uint32_t bytes;
        uint32_t syn_flows;
        uint32_t icmp_reply;
        uint32_t icmp_request;
        uint32_t udp_flows;
        double port_80_flows;
        double port_80_packets;
        double port_80_flows_packets;
        int suspicion;
} hash_table_record_t;
```

Obrázek 5.2: Struktura záznamu v hash tabulce.

Velkou nevýhodou sondy FlowMon je zatím neexistující rozhraní pro přímou práci se záznamy v paměti NetFlow cache. Pro detekci založenou na sledování heuristik je tedy potřeba vytvořit si vlastní paměť, kam se budou jednotlivé heuristiky ukládat. K tomu posloužila implementace hash tabulky GHashTable z knihovny GLib. Knihovna poskytuje poměrně intuitivní rozhraní pro práci s hash tabulkou a poskytuje všechny funkce, které byly pro tyto účely nezbytné. Struktura záznamu v hash tabulce je uvedena na obr. 5.2.

Při startu pluginu se vždy nejprve provede inicializační funkce všech dílčích pluginů. V procesním jde konkrétně o funkci `plugin_process_init`, která alokuje místo v paměti pro privátní strukturu procesního pluginu `process_private_t` a inicializuje všechny její hodnoty. Tato struktura je poté přístupná ve všech základních funkcích v rámci procesního pluginu a využívám ji pro uložení seznamu útočných adres, hash tabulky a proměnných, které sledují počet paketů, bytů a toků na globální úrovni.

Architektura pluginu také podporuje připojení dalších libovolných hodnot do NetFlow záznamu. K tomu slouží struktura `process_record_t`, která dané hodnoty uloží s určitým offsetem k NetFlow záznamu do paměti a ty pak jsou přístupné pomocí tzv. getterů. Jde o velmi jednoduchý způsob, jak modifikovat NetFlow záznamy informacemi, které v něm normálně chybí a usnadnit si práci. Tuto možnost jsem využil pro uložení předchozí hodnoty bytů a paketů v toku, abych při modifikaci toku dokázal vypočítat, kolik bytů a paketů přibylo a přičíst je do patřičného počitadla. Diagram běhu pluginu je zobrazen na obrázku 5.3. Modře jsou zvýrazněny standardní funkce a součásti FlowMon exportéru, jejichž definice je ale na programátorovi. Černě jsou zvýrazněny dodané funkce, které slouží pro sbírání a vyhodnocování heuristik. Flowmonexp značí exportér, který řídí celý běh pluginu.



Obrázek 5.3: Diagram běhu pluginu.

Po příchodu prvního paketu toku se vytvoří záznam v paměti NetFlow cache a je zavolána funkce `plugin_process_create`, která umožňuje pracovat s právě vytvořeným tokem. V této funkci začíná sběr všech dat, která jsou později využita jako heuristiky při hodnocení záznamů v hash tabulce. Při příchodu dalšího paketu toku přijde na řadu funkce `plugin_process_update`, kde se pokračuje ve sběru dat. Přírůstek v počtu paketů a bytů je zde vypočten odečtením předchozí hodnoty, která je uložená v uživatelské části NetFlow záznamu, od aktuální hodnoty. Na konci těchto funkcí se volá funkce `evaluate_window`, která hlídá, jestli již neuplynulo časové okno (1 minuta). Pokud ano, projde hash tabulku a každý její záznam je ohodnocen hodnotící funkcí `evaluate_record`. Ta vyhodnotí všechny heuristiky uložené v záznamu, a pokud je úroveň podezření větší nebo rovna 2, záznam je zkopírován do lineárního seznamu black_list. Po dokončení průchodu hash tabulkou je black_list vytisknut do logovacího souboru funkcí `to_log` a všechny prvky black_listu a hash tabulky jsou uvolněny z paměti. Do dalšího časového okna tedy nepřecházejí žádné informace, což značně šetří paměť. Při expiraci záznamu z NetFlow cache je volána funkce

`plugin_process_release`, která sbírá informace o již kompletních tocích, což je užitečné zejména pro detekci útoků SYN flood, které se projevují jednopaketovými toky s příznakem SYN.

Sběr samotných dat do hash tabulky provádí funkce `save_ip_info`. Podle příznaku, se kterým je zavolána, uloží data do příslušného záznamu v hash tabulce, který je adresován hash hodnotou ze zdrojové IP adresy. Funkce `evaluate_record` vyhodnocuje heuristiky na základě prahových hodnot, které byly stanoveny empiricky. Nevýhodou tohoto přístupu je, že na každé síti bude nutné nakonfigurovat prahové hodnoty a provést krátké testování, aby se zajistil minimální počet falešných poplachů. Tomuto by se dalo vyhnout implementací fáze učení, která by sama během prvního spuštění sledovala provoz na dané síti a získala potřebné prahové hodnoty. Právě to by mohlo být velmi zajímavé rozšíření.

5.4 Výstup pluginu

Jelikož je plugin konzolová aplikace, výstupy jsou realizovány v podobě logovacího souboru, kam se zaznamenávají všechny podezřelé IP adresy, které byly v daném časovém okně detekovány. Z tohoto souboru lze jednoduše zjistit, jak dlouho útok přibližně trval, které adresy se ho účastnily a kolik přenesly dat. Na tento výstup by při provozu mohl reagovat jiný systém, který by na základě analýzy logovacího souboru upozornil správce sítě nebo dále analyzoval zalogovaná data. Ukázka výstupu pluginu je uvedena na obrázku 5.4.

```
================================================================================================
[Start: 04/18/13 15:45:57 --> End: 04/18/13 15:46:57]
================================================================================================
[  1] SOURCE IP:    10.10.10.229  PACKETS:   316162  BYTES: 4287583135  FLOWS:   316148  SUSPICION:  2

================================================================================================
[Start: 04/18/13 15:46:57 --> End: 04/18/13 15:47:57]
================================================================================================
[  1] SOURCE IP:    10.10.10.229  PACKETS:   356695  BYTES: 4289955833  FLOWS:
356695  SUSPICION:  2
```

Obrázek 5.4: Ukázka výstupu pluginu.

Kapitola 6

Analýza útoků a testování

Při implementaci nástroje pro detekci rozsáhlých útoků jako DoS a DDoS je velmi důležité provést důkladné testování výkonnosti a spolehlivosti. Plugin nesmí zabírat příliš mnoho výpočetní a paměťové kapacity sondy, aby i při útoku byla sonda schopna bezproblémově vykonávat svou standardní činnost. Již od začátku jsem se tedy při implementaci snažil dbát na výkon pluginu, který by neměl mít problémy pracovat i na velmi vytížených sítích. V této kapitole budou prezentovány informace získané analýzou jednotlivých typů útoků a také výsledky, kterých plugin dosáhl při praktickém testování detekce těchto útoků.

Analýza provozu generovaného útočnými programy je prováděna pomocí programu Wireshark pro zachytávání paketů a k útokům jsou využity nástroje HPing (SYN, ICMP) a LOIC (HTTP, UDP, TCP). IP adresa 192.168.1.2 představuje útočící počítač a adresa 192.168.1.1 cíl útoku. Naměřené hodnoty se v intervalech 1 sekundy ukládají do souboru, který je zpracován skriptem programu Gnuplot a automaticky vygeneruje grafy počtu paketů, bytů a toků.

6.1 Testovací metodika

Součástí detekce je také návrh testovací metodiky, která by dokázala spolehlivě ověřit funkčnost pluginu před jeho nasazením. Pokud se bavíme o detekci útoků na úrovni kolektoru, zde je testování poměrně přímočaré. K dispozici je prakticky neomezené množství NetFlow dat a to i s útoky, které se dají jednoduše zpracovat pomocí skriptů. Testování na úrovni exportéru však vyžaduje živý provoz, který by monitorovací rozhraní sondy zachytilo. Většina druhů detekce totiž sleduje určité vlastnosti za daný čas, jako počet paketů za sekundu, a tato informace by při načítání dat ze souboru neměla žádnou hodnotu.

Ideálním řešením by v tomto případě bylo nasadit plugin na síti s dostatečným provozem a provést rozsáhlý útok. Prakticky ale samozřejmě není možné spustit testovací útok na síť, kudy prochází provoz běžných uživatelů. Při testování proto využívám nástroj `Tcpreplay`, který dokáže volitelnou rychlostí přehrávat síťový provoz uložený v PCAP formátu na dané rozhraní. Nezůstávají však zachovány časové rozestupy mezi jednotlivými pakety, takže chování přehraného provozu není úplně stejné jako originálu.

Samotné testování se skládá ze dvou části. Nejprve je plugin testován v laboratorních podmínkách s přehrávaným provozem a simulovanými útoky, což by mělo ověřit, zda je opravdu schopen detekovat uvažované útoky. V druhé části je plugin nasazen na reálnou a velmi vytíženou síť, která je monitorována sondou FlowMon. Tento test by měl posloužit k otestování výkonnosti a spolehlivosti. Konkrétně jde o síť areálu Listovy koleje. Zaznamе-

naný a anonymizovaný provoz z této sítě je použit pro testování v laboratorních podmínkách a je přehráván na sondu nástrojem `Tcpreplay`. K útokům jsou využity nástroje LOIC a HPing, které jsou detailně popsány v kapitole 2. Délka všech testů v laboratorních podmínkách je přibližně 15-25 minut.

Během testování byly prahové hodnoty jednotlivých metrik stanoveny na hodnoty, které jsou uvedeny v tabulce 6.1. Určení prahových hodnot důkladným testováním snižuje počet falešných poplachů.

Heuristika	**Prahová hodnota**	**Body**
Min. počet paketů	5 000	0
Poměr TCP pakety/pakety	0.913	1
Poměr UDP pakety/pakety	0.35	1
Poměr ICMP pakety/pakety	0.2	1
Poměr toky/pakety (port 80)	0.002	1
UDP toky a ICMP Dst. U. pakety	UDP = 5 000, ICMP = 50	1
Jednopaketové SYN toky	6 000	2
Echo Request pakety	6 000	1

Tabulka 6.1: Prahové hodnoty heuristik.

6.2 SYN flood

Příklad spuštění nástroje HPing: `sudo hping3 --flood -S 192.168.1.1`

-flood: Určuje interval mezi odesíláním jednotlivých paketů. V tomto případě se pakety odesílají nejvyšší možnou rychlostí.

-S: Říká, aby HPing odesílal TCP pakety s příznakem SYN.

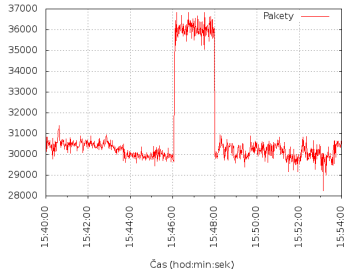
V tabulce 6.2 je zobrazena jedna iterace útočného provozu, který je generován nástrojem HPing. Jde vidět, že útočník neustále zasílá pakety SYN a ignoruje odpovědi cíle útoku. Vzniká tedy velké množství toků s jedním paketem a příznakem SYN.

No.	**Time**	**Source**	**Destination**	**Proto.**	**Len.**	**Info**
1	0.000000	192.168.1.2	192.168.1.1	TCP	60	dca » http [SYN]
2	0.000045	192.168.1.1	192.168.1.2	TCP	58	http » dca [SYN, ACK]

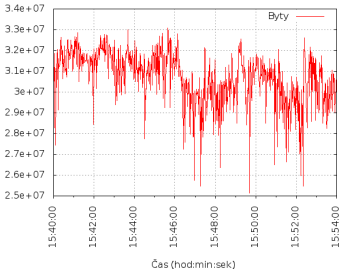
Tabulka 6.2: Výpis provozu SYN flood útoku.

V průběhu testování pluginu byl simulován jeden SYN flood útok pomocí nástroje HPing. Na obrázcích 6.1 až 6.3 lze vidět, že útok trval 2 minuty a nejvíce se projevil v počtu toků a paketů. Větší verze těchto grafů jsou k dispozici v příloze C.

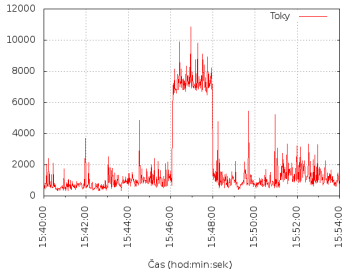
Útok byl detekován v celkem třech časových oknech (1 minuta) za sebou. Přesah do třetí minuty je způsoben tím, že jednopaketové toky s příznakem SYN jsou detekovány až při expiraci a některé z nich expirují až po zastavení útoku. Během této doby zaznamenal plugin dohromady přesně 679 360 jednopaketových toků s příznakem SYN z IP adresy 10.10.10.229 a úspěšně tento útok detekoval. Prahová hodnota byla nastavena na maximálně 6 000 těchto toků. Výstup pluginu je uveden na obrázku 6.4.



Obrázek 6.1: Počet paketů.



Obrázek 6.2: Počet bytů.



Obrázek 6.3: Počet toků.

```
================================================================================================
[Start: 04/18/13 15:45:57 --> End: 04/18/13 15:46:57]
================================================================================================
[  1] SOURCE IP:    10.10.10.229  PACKETS:   316162  BYTES: 4287583135  FLOWS:   316148  SUSPICION:  2

================================================================================================
[Start: 04/18/13 15:46:57 --> End: 04/18/13 15:47:57]
================================================================================================
[  1] SOURCE IP:    10.10.10.229  PACKETS:   356695  BYTES: 4289955833  FLOWS:   356695  SUSPICION:  2

================================================================================================
[Start: 04/18/13 15:47:57 --> End: 04/18/13 15:48:57]
================================================================================================
[  1] SOURCE IP:    10.10.10.229  PACKETS:     6541  BYTES: 4294861725  FLOWS:     6541  SUSPICION:  2
```

Obrázek 6.4: Výstup pluginu při útoku SYN flood.

Použitá prahová hodnota jednopaketových toků s příznakem SYN se ukázala být dostačující. Je sice velmi nepravděpodobné, že by neútočná IP adresa generovala větší množství takovýchto toků, ale takto nastavenou prahovou hodnotou se vyloučí malé útoky, které nejsou pro infrastrukturu nebezpečné. Pro detekci menších útoků stačí prahovou hodnotu pouze snížit.

6.3 ICMP flood

Příklad spuštění nástroje HPing:

-flood: Určuje interval mezi odesíláním jednotlivých paketů. V tomto případě se pakety odesílají nejvyšší možnou rychlostí.

-icmp: Říká, aby HPing odesílal ICMP pakety. Jejich typ je ve výchozím stavu Echo Request.

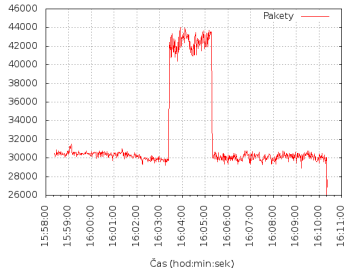
Tabulka 6.3 obsahuje krátkou část ICMP flood útoku. V tuto chvíli by však mohlo jít o běžné použití příkazu ping, a proto je nutné nastavit prahovou hodnotu pro každou IP adresu, která stanoví, kolik Echo Request paketů je přijatelných za určitý čas.

No.	**Time**	**Source**	**Destination**	**Proto.**	**Len.**	**Info**
1	0.000413	192.168.1.2	192.168.1.1	ICMP	60	Echo (ping) request
2	0.000459	192.168.1.2	192.168.1.1	ICMP	42	Echo (ping) reply

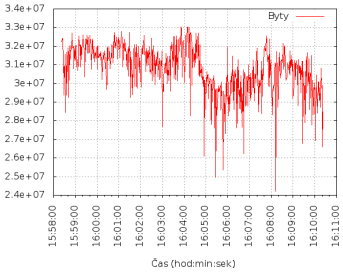
Tabulka 6.3: Výpis provozu ICMP flood útoku.

Při tomto testu proběhl jeden ICMP flood útok pomocí nástroje HPing, který trval opět přibližně 2 minuty. Útok se velmi razantně projevil v počtu paketů a nárazově také v počtu

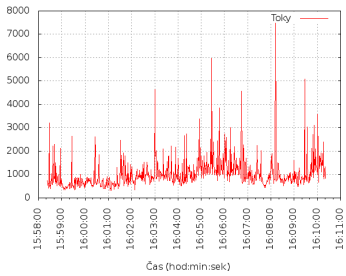
toků. Všechny útočné pakety patřily do těchto toků a nové by vznikly až po jejich expiraci. I když útok probíhal z jedné adresy, nástroj HPing útočí na náhodná čísla portů, pokud se nezadá konkrétní port, a proto vzniklo v počáteční fázi útoku větší množství toků. Průběh útoku je zobrazen na obrázcích 6.5 až 6.7.



Obrázek 6.5: Počet paketů.



Obrázek 6.6: Počet bytů.



Obrázek 6.7: Počet toků.

Plugin úspěšně detekoval útok z IP adresy 10.10.10.229 probíhající v čase 16:03:21 až 16:05:21, což odpovídá i době zvýšeného množství paketů na grafu 6.5. Za tuto dobu odeslal útočník 1 434 259 ICMP Echo Request požadavků. ICMP provoz dosáhl 28.5 % z celkového provozu a útok byl úspěšně detekován.

```
================================================================================================
[Start: 04/18/13 16:03:21 --> End: 04/18/13 16:04:21]
================================================================================================
[  1] SOURCE IP:  10.10.10.229  PACKETS: 3840274007  BYTES:   10014152  FLOWS:   357693  SUSPICION:  2

================================================================================================
[Start: 04/18/13 16:04:21 --> End: 04/18/13 16:05:21]
================================================================================================
[  1] SOURCE IP:  10.10.10.229  PACKETS: 4037690661  BYTES:   10058637  FLOWS:   359237  SUSPICION:  2
```

Obrázek 6.8: Výstup pluginu při útoku ICMP flood.

6.4 HTTP flood

Program LOIC byl spuštěn s následujícím nastavením:

- **Cíl útoku**: 192.168.1.1, port: 80
- **Počet vláken**: 5
- **Metoda**: HTTP

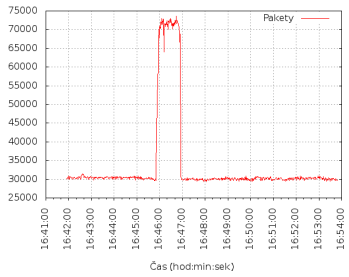
V tabulce 6.4 je zobrazena komunikace jednoho z vláken aplikace LOIC a to konkrétně vlákna využívajícího zdrojový port 51013. Každé z pěti vláken má svůj vlastní zdrojový port a vytváří samostatný tok. Vlákno nejprve standardním způsobem naváže úplnou TCP komunikaci a poté zasílá velké množství HTTP GET nebo POST požadavků cílovému serveru. V rámci toku je odesíláno velké množství HTTP požadavků a tedy paketů. Tato heuristika je velmi užitečná při samotné detekci.

HTTP flood útok nástrojem LOIC byl spuštěn přibližně 1 minutu. Šlo tedy o velmi nárazový útok, který se razantně projevil v počtu paketů a následně velmi rychle skončil.

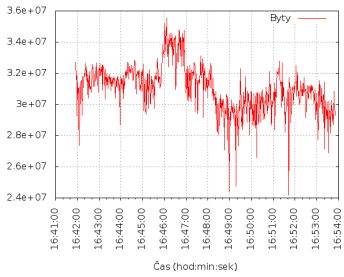
Útok byl detekován v čase 16:45:49 až 16:46:49, kdy TCP provoz dosáhl hodnoty 92.4 % a poměr mezi přijatými toky a pakety na port 80 z této IP adresy byl stanoven na hodnotu 0.00005, což je hluboko pod prahovou hodnotou. Úroveň podezření byla zvýšena na 2.

No.	Time	Source	Destination	Proto.	Len.	Info
3	0.000411	192.168.1.2	192.168.1.1	TCP	74	51013 » http [SYN]
4	0.000456	192.168.1.1	192.168.1.2	TCP	74	http » 51013 [SYN, ACK]
9	0.000606	192.168.1.2	192.168.1.1	TCP	66	51013 » http [ACK]
21	0.001947	192.168.1.2	192.168.1.1	HTTP	83	GET / HTTP/1.0
22	0.001961	192.168.1.1	192.168.1.2	TCP	66	http » 51013 [ACK]
...						
339	0.200954	192.168.1.1	192.168.1.2	TCP	66	http » 51013 [FIN, ACK]
341	0.201169	192.168.1.2	192.168.1.1	TCP	66	51013 » http [ACK]

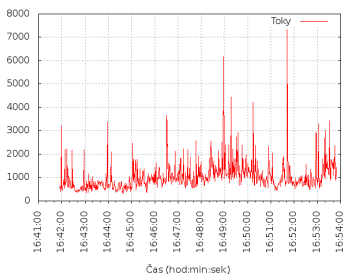
Tabulka 6.4: Výpis provozu HTTP flood útoku.



Obrázek 6.9: Počet paketů.



Obrázek 6.10: Počet bytů.



Obrázek 6.11: Počet toků.

```
================================================================================================
[Start: 04/18/13 16:45:49 --> End: 04/18/13 16:46:49]
================================================================================================
[  1] SOURCE IP:    10.10.10.229  PACKETS: 2594318253  BYTES:   97520857  FLOWS:  1156825  SUSPICION:  2
```

Obrázek 6.12: Výstup pluginu při útoku HTTP flood.

6.5 UDP flood

Program LOIC byl spuštěn s následujícím nastavením:

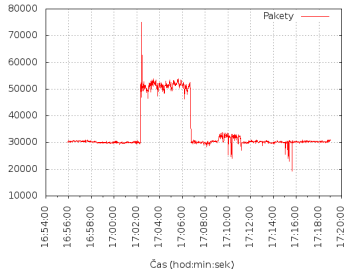
- **Cíl útoku**: 192.168.1.1, port: 80
- **Počet vláken**: 5
- **Metoda**: UDP

Tabulka 6.5 ukazuje, že každé z pěti vláken využívá specifický port a poté, co všech pět vláken odešle své požadavky, začíná opět první vlákno na stejném portu. V rámci UDP toku se tedy objeví velké množství paketů.

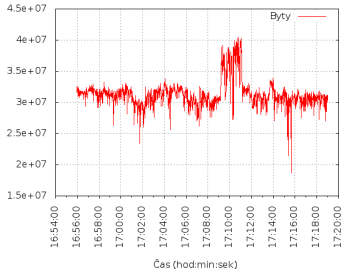
No.	Time	Source	Destination	Proto.	Len.	Info
1	0.000809	192.168.1.2	192.168.1.1	UDP	74	Src port: 39373 Dst port: http
2	0.000925	192.168.1.2	192.168.1.1	UDP	74	Src port: 58496 Dst port: http
3	0.001066	192.168.1.2	192.168.1.1	UDP	74	Src port: 55884 Dst port: http
4	0.001210	192.168.1.2	192.168.1.1	UDP	74	Src port: 45382 Dst port: http
5	0.015288	192.168.1.2	192.168.1.1	UDP	74	Src port: 33051 Dst port: http
6	0.015396	192.168.1.2	192.168.1.1	UDP	74	Src port: 39373 Dst port: http

Tabulka 6.5: Výpis provozu UDP flood útoku.

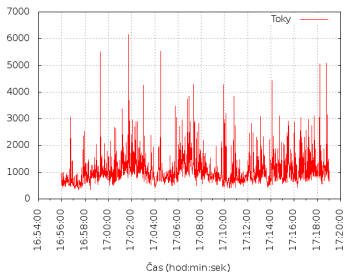
Útok UDP flood byl generován nástrojem LOIC na port 80 a projevil se zejména zvýšeným počtem paketů a také bytů.



Obrázek 6.13: Počet paketů.



Obrázek 6.14: Počet bytů.



Obrázek 6.15: Počet toků.

Plugin odhalil útok probíhající v době 17:01:55 až 17:05:55, kdy UDP provoz zabíral skoro 50 % z celkového provozu, což je velmi netypické a tato hodnota překračuje stanovenou prahovou hodnotu 35 %. Za normálních okolností byla hladina UDP provozu okolo 15 %. Poměr mezi pakety a toky na portu 80 byl v první minutě útoku stanoven na 0.00004 a následující 4 minuty nevznikly žádné nové toky z této IP adresy. V rámci těchto toků však bylo odesláno abnormálně velké množství paketů, což vzhledem k nulovému počtu vzniklých toků z této adresy přispělo ke zvýšení podezření na hodnotu 2.

```
================================================================================================
[Start: 04/18/13 17:01:55 --> End: 04/18/13 17:02:55]
================================================================================================
[  1] SOURCE IP:    10.10.10.229  PACKETS: 1582054836  BYTES:   46869678  FLOWS:   781205  SUSPICION:  2

================================================================================================
[Start: 04/18/13 17:02:55 --> End: 04/18/13 17:03:55]
================================================================================================
[  1] SOURCE IP:    10.10.10.229  PACKETS:  437666008  BYTES:   76737620  FLOWS:  1278967  SUSPICION:  2

================================================================================================
[Start: 04/18/13 17:03:55 --> End: 04/18/13 17:04:55]
================================================================================================
[  1] SOURCE IP:    10.10.10.229  PACKETS:  134156666  BYTES:   77279670  FLOWS:  1287995  SUSPICION:  2

================================================================================================
[Start: 04/18/13 17:04:55 --> End: 04/18/13 17:05:55]
================================================================================================
[  1] SOURCE IP:    10.10.10.229  PACKETS: 3454855262  BYTES:   78212604  FLOWS:  1303659  SUSPICION:  2

================================================================================================
[Start: 04/18/13 17:05:55 --> End: 04/18/13 17:06:55]
================================================================================================
[  1] SOURCE IP:    10.10.10.229  PACKETS: 2252496986  BYTES:   62207828  FLOWS:  1036798  SUSPICION:  2
```

Obrázek 6.16: Výstup pluginu při útoku UDP flood.

6.6 TCP flood

Program LOIC byl spuštěn s následujícím nastavením:

- **Cíl útoku**: 192.168.1.1, port: 80
- **Počet vláken**: 5
- **Metoda**: TCP

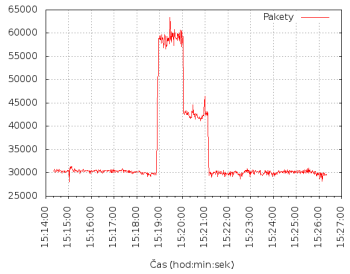
Podle informací v tabulce 6.6 můžeme usoudit, že LOIC při útoku TCP flood zahajuje úplnou TCP komunikaci a poté odesílá obrovské množství požadavků na cílový server v

rámci jednoho toku, dokud není útok ukončen nebo nevyprší aktivní časovač spojení. V tomto případě byl útok ukončen a spojení bylo řádně ukončeno.

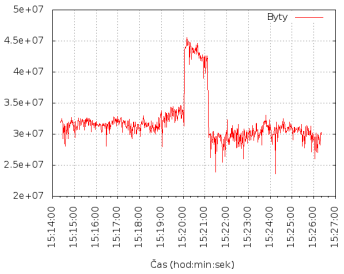
No.	Time	Source	Destination	Proto.	Len.	Info
1	0.000000	192.168.1.2	192.168.1.1	TCP	74	51242 » http [SYN]
2	0.000028	192.168.1.1	192.168.1.2	TCP	74	http » 51242 [SYN, ACK]
3	0.000376	192.168.1.2	192.168.1.1	TCP	66	51242 » http [ACK]
6	0.000384	192.168.1.2	192.168.1.1	HTTP	98	Continuation or non-HTTP traffic
7	0.000401	192.168.1.1	192.168.1.2	TCP	66	http » 51242 [ACK]
...						
4015	6.012714	192.168.1.1	192.168.1.2	TCP	66	http » 51242 [FIN, ACK]
4016	6.013113	192.168.1.2	192.168.1.1	TCP	66	51242 » http [ACK]

Tabulka 6.6: Výpis provozu TCP flood útoku.

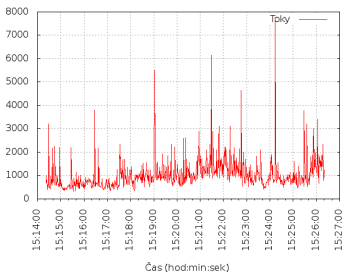
TCP flood útok byl opět simulován nástrojem LOIC na port 80. Z výsledných grafů lze vidět, že se projevil zejména v počtu přenesených bytů a paketů. Přibližně v polovině útoku byla jeho intenzita snížena na polovinu.



Obrázek 6.17: Počet paketů.



Obrázek 6.18: Počet bytů.



Obrázek 6.19: Počet toků.

Výstup pluginu ukazuje, že byl odhalen útok v období 15:18:20 až 15:20:20 z IP adresy 10.10.10.229. TCP provoz v této době zabíral přes 92 % z celkového provozu a překročil prahovou hodnotu 91 %. Poměr mezi toky a pakety na port 80 byl stanoven na 0.00005. Tato IP adresa tedy dosáhla celkového podezření 2.

```
==================================================================================================
[Start: 04/18/13 15:18:20 --> End: 04/18/13 15:19:20]
==================================================================================================
[  1] SOURCE IP:    10.10.10.229  PACKETS: 3612107892  BYTES:   32619536  FLOWS:   379527  SUSPICION:  4

==================================================================================================
[Start: 04/18/13 15:19:20 --> End: 04/18/13 15:20:20]
==================================================================================================
[  1] SOURCE IP:    10.10.10.229  PACKETS: 3583567551  BYTES:  269092439  FLOWS:   761682  SUSPICION:  4
```

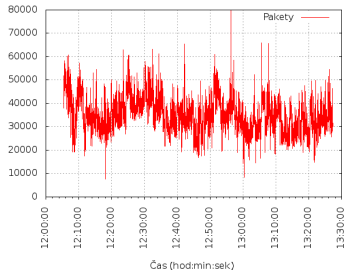
Obrázek 6.20: Výstup pluginu při útoku TCP flood.

6.7 Testování na reálné síti

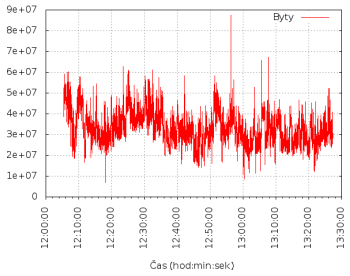
V reálném provozu musí plugin splňovat požadavky zejména na výkonnost, nenáročnost na výpočetní zdroje a generovat co nejmenší množství falešných poplachů. K tomuto testu byla využita síť areálu Listovy koleje, kde byl plugin spuštěn po dobu přibližně 80 minut. Během tohoto testu byly zjištěny následující informace:

- Průměrný počet záznamů v hash tabulce se pohyboval okolo 8 000.
- Po celou dobu testu se neobjevila žádná podezřelá IP adresa.
- Vytížení sondy nepřesáhlo 3 %.
- Plugin využíval 1 až 2 % z 2.9 GB dostupné RAM paměti.
- Průměrné zastoupení jednotlivých transportních protokolů bylo následující: TCP = 72 až 80 %, UDP = 11 až 28 %, ICMP = 0.01 až 0.07 %

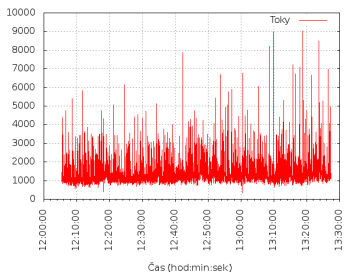
Množství provozu, které za tuto dobu sondou prošlo, je zobrazeno na obrázcích 6.21 až 6.23 v 1 sekundových intervalech.



Obrázek 6.21: Počet paketů.



Obrázek 6.22: Počet bytů.



Obrázek 6.23: Počet toků.

6.8 Shrnutí výsledků testování

Výsledky testů na síti Listovy koleje dokazují, že plugin je připraven na provoz na rozsáhlé síti a při zvolení vhodných prahových hodnot nedochází ke generování falešných poplachů. Zároveň také využívá minimum výpočetního výkonu samotné sondy.

Testování v laboratorních podmínkách ukázalo, že plugin dokáže spolehlivě detekovat uvažované útoky DoS, které významně zvýší poměr přenášených dat daného transportního protokolu. Útok DDoS je odlišitelný jednoduše tak, že v logovacím souboru bude zapsáno pro dané okno velké množství podezřelých IP adres.

Byly tedy splněny požadavky pro provoz na reálné síti a plugin úspěšně detekoval uvažované útoky. Detekce dalších typů útoků je otázkou pouze přidání sběru vhodných heuristik, určení prahové hodnoty a bodového ohodnocení.

6.9 Možná rozšíření

Plugin je schopen provádět spolehlivou detekci útoků DoS a DDoS, ovšem existuje několik oblastí, kde by bylo možné implementovat rozšíření, která by usnadnila jeho provozování a rozšířila množinu detekovaných útoků.

Asi nejpodstatnějším rozšířením je automatické stanovení prahových hodnot při prvním spuštění pluginu, tzv. fáze učení. Ta by zjednodušila prvotní nastavení pluginu při jeho nasazení i méně zkušeným zákazníkům.

Obecně může být plugin neustále rozšiřován o detekci různých typů útoků. V tuto chvíli se zaměřuje zejména na útoky proti webovým serverům různými transportními protokoly

a útoky UDP flood, SYN flood a ICMP flood. Do budoucna by ale mohl pokrýt ještě větší množství útoků.

Samotný logovací soubor se přímo nabízí pro další zpracování a vyvození souvislostí mezi jednotlivými IP adresami jako určení sítě, ze které pochází více útočných IP adres, či počet IP adres, které se účastnily konkrétního útoku a podobně.

Kapitola 7

Závěr

V teoretické části této bakalářské práce jsem se seznámil s běžnými útoky DoS a DDoS a jejich charakteristikami. Dále jsem prostudoval technologii NetFlow a prostředí tvorby pluginů pro sondu FlowMon. Po návrhu několika možných řešení jsem vzal v potaz zjištěná omezení sondy FlowMon a navrhnul systém, který na základě několika heuristik bodově hodnotí provoz všech zdrojových IP adres.

V praktické části jsem na základě zjištěných poznatků implementoval plugin pro exportér sondy FlowMon, který dokáže detekovat vybrané útoky DoS a DDoS. Pro testování jsem využil zaznamenaný anonymizovaný provoz ze sítě areálu Listovy koleje, který jsem v laboratoři přehrával proti hardwarové sondě FlowMon a zároveň simuloval útoky. Poslední částí testování bylo nasazení na reálnou síť. V rámci provedených testů uspěl plugin jak v laboratorních podmínkách, tak na reálné síti a prokázal dostatečnou výkonnost a spolehlivost pro praktické nasazení.

Stále však samozřejmě existuje několik oblastí, kde by bylo vhodné implementovat různá rozšíření. Ta jsou popsána v kapitole 6 a týkají se zejména zjednodušení práce se sondou a přidání dalších informací o útocích.

Během tvorby pluginu jsem získal detailní znalosti o útocích typu DoS a DDoS a také znalosti o praktickém nasazení technologie NetFlow. Seznámil jsem se s konkrétním využitím NetFlow na sondě FlowMon a strukturou exportéru, který je na této sondě implementován. Velmi zajímavé bylo také zaměřit se při vývoji na výkon pluginu a následné ladění pluginu, aby bylo možné ho provozovat na velmi vytížených sítích.

Útoky DoS a DDoS jsou stále velmi aktuálním tématem, a proto doufám, že se plugin ukáže být užitečný a použitelný při reálném nasazení pro zákazníky společnosti INVEA-TECH. Rád bych se také vývoji a rozšiřování tohoto pluginu věnoval i nadále a implementoval některá navrhnutá rozšíření.

S touto prací jsem se zúčastnil soutěže studentské tvůrčí činnosti EEICT, kterou již tradičně pořádají fakulty FIT a FEKT. Po prezentaci dosažených výsledků byla tato práce oceněna 2. místem v sekci Informační systémy bakalářské formy studia.

Literatura

[1] Cisco: Introduction to Cisco IOS NetFlow. [online], květen 2012, [cit. 2013-04-24].
URL http://www.cisco.com/en/US/prod/collateral/iosswrel/ps6537/ps6555/ps6601/prod_white_paper0900aecd80406232.html

[2] Cisco: NetFlow Services Solutions Guide. [online], září 2001, [cit. 2013-01-20].
URL http://www.cisco.com/en/US/products/sw/netmgtsw/ps1964/products_implementation_design_guide09186a00800d6a11.html

[3] Claise, B.: Specification of the IP Flow Information Export (IPFIX) Protocol. RFC 5101. leden 2008.

[4] Claise, B.: Cisco Systems NetFlow Services Export Version 9. RFC 3954. říjen 2004.

[5] Dobbins, R.; Morales, C.: Worldwide Infrastructure Security Report. [online], 2011, [cit. 2013-01-20].
URL http://ddos.arbornetworks.com/report/

[6] Eddy, W.: TCP SYN Flooding Attacks and Common Mitigations. RFC 4987. srpen 2007.

[7] Fajmon, B.; Koláček, J.: Pravděpodobnost, statistika a operační výzkum. Elektronické skriptum FEKT VUT,. 2005, [cit. 2013-04-24].

[8] INVEA-TECH: FlowMon. [online], [cit. 2013-01-20].
URL http://www.invea.cz/products/flowmon

[9] INVEA-TECH: FlowMon sondy. [online], [cit. 2013-01-20].
URL http://www.invea.cz/produkty-sluzby/flowmon/flowmon-sondy

[10] Patrikakis, C.; Masikos, M.; Zouraraki, O.: Distributed Denial of Service Attacks. [online], prosinec 2004, [cit. 2013-01-20].
URL http://www.cisco.com/web/about/ac123/ac147/archived_issues/ipj_7-4/dos_attacks.html

[11] Plch, M.: Detekce DoS útoků pomocí analýzy síťových toků: Diplomová práce. Praha: ČVUT v Praze, Fakulta informačních technologií. květen 2012, [cit. 2013-04-24].

[12] Postel, J.: Transmission Control Protocol. RFC 793. září 1981.

[13] Prolexic: Q1 2013 Global DDoS Attack Report. [online], 2013, [cit. 2013-04-24].
URL www.prolexic.com/knowledge-center-ddos-attack-report-2013-q1.html

[14] Raghavan, S. V.; Dawson, E.: *An Investigation into the Detection and Mitigation of Denial of Service (DoS) Attacks*. Springer, 2011, ISBN 978-81-322-0277-6.

[15] RSnake: Slowloris HTTP DoS. [online], [cit. 2013-01-20].
URL http://ha.ckers.org/slowloris/

[16] Rylich, J.: SOPA, PIPA & ACTA aneb Boj o svobodu na Internetu. [online], 2012, roč. 16, č. 2, [cit. 2013-05-07].
URL
http://www.ikaros.cz/sopa-pipa-acta-aneb-boj-o-svobodu-na-internetu

[17] Sanfilippo, S.: HPing. [online], 2006, [cit. 2013-04-24].
URL http://www.hping.org/

[18] Slížek, D.; Vyleťal, M.: DDoS útok zasáhl weby mobilních operátorů. [online], březen 2013, [cit. 2013-04-24].
URL http://www.lupa.cz/clanky/
ddos-pokracuje-cilem-jsou-dnes-weby-mobilnich-operatoru/

[19] Verma, D.: LOIC (Low Orbit Ion Cannon) - DOS attacking tool. [online], prosinec 2011, [cit. 2013-04-24].
URL http://resources.infosecinstitute.com/loic-dos-attacking-tool/

Příloha A

Obsah DVD

- `README.txt` ..... manuál k použití
- `BP_Jan_Hunka_2013.pdf` ..... text práce ve formátu PDF
- `ddos_detector` ..... zdrojové soubory pluginu
  - `input` ..... zdrojové soubory vstupního pluginu
  - `process` ..... zdrojové soubory procesního pluginu
    - `ddos.h` ..... knihovna s operacemi jednosměrného seznamu
  - `export` ..... zdrojové soubory exportního pluginu
  - `Makefile` ..... makefile pro všechny pluginy
- `flowmonexp` ..... knihovny pro vývoj pluginů sondy FlowMon
- `tex` ..... zdrojové soubory textové části BP
  - `img` ..... obrázky a grafy použité v textové části BP

Příloha B

Manuál

Překlad pluginu vyžaduje virtuální nebo hardwarovou sondu FlowMon s exportérem ve verzi 3.02.05 nebo vyšší. Dále jsou nutné knihovny pro tvorbu pluginů na exportéru (složka flowmonexp) a knihovna GLib pro implementaci hash tabulky. Knihovna GLib bývá na sondách FlowMon již obsažena a není ji potřeba zvlášť dodávat. Plugin tyto knihovny čerpá ze standardního umístění knihoven na sondě, tj. adresář /usr/include/. Ke spuštění pluginu ddos je potřeba načíst tři dílčí pluginy:

- **ddos-input**: Zpracovává pakety ze zadaného rozhraní a vytváří z nich NetFlow záznamy.
- **ddos-process**: Detekuje ze vznikajících toků útoky DoS a DDoS.
- **stdout**: V případě potřeby může vytisknout exportované toky na standardní výstup. Je nutný ke spuštění pluginu.

Nejprve je nutné spustit překlad všech částí pluginu příkazem make na úrovni složky ddos_detector. Zde přítomný makefile zajistí přeložení všech zdrojových souborů. Pro samotnou detekci na monitorovacím rozhraní eth1 bude spuštění pluginu ze složky ddos_detector vypadat následovně:

```
sudo flowmonexp -X ./input/input_ddos.so -X ./process/process_ddos.so
-X ./export/stdout.so -I input-ddos:eth1 -P process-ddos: -E stdout:
```

Pokud jde o spuštění na reálné síti, plugin by v tomto bodě měl již fungovat a v minutových intervalech provádět kontrolu dat z hash tabulky. V případě, že jde o testování v laboratorních podmínkách, je potřeba přehrávat po stejném segmentu sítě, kde se nachází sonda, zaznamenaný provoz v podobě souborů PCAP pomocí nástroje Tcpreplay. Monitorovací rozhraní sondy tento provoz zaznamená a plugin ho zpracuje pro detekci. Útoky lze simulovat nástroji LOIC či HPing na adresu administrativního rozhraní sondy. Oba nástroje jsou volně dostupné. Nástroj HPing se dá získat jednoduše příkazem:

```
sudo apt-get install hping3
```

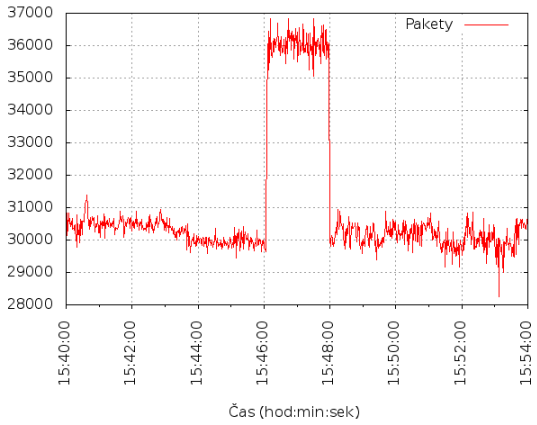
Program LOIC je implementován v jazyce C#, a proto je nutné pro jeho spuštění v Unixu využít prostředí Mono. Soubor je dostupný na http://sourceforge.net/projects/loic/.

Všechny prahové hodnoty, které jsou v pluginu využity, se dají nastavit v hlavičce souboru ddos-process.c. Soubor log.txt se po spuštění nachází v hlavní složce programu.

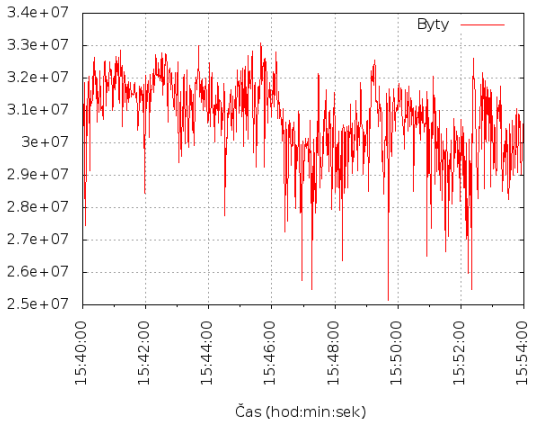
Příloha C

Grafy útoků

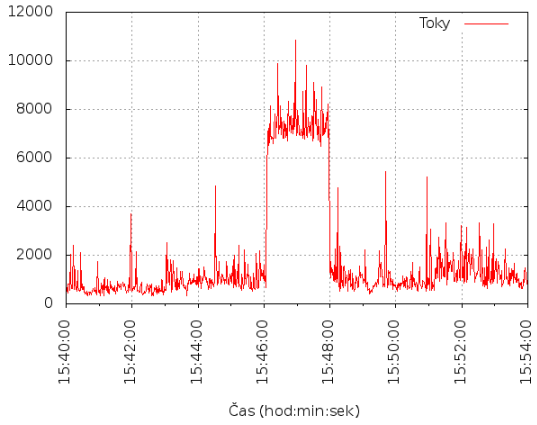
C.1 Útok SYN flood



Obrázek C.1: Počet paketů.

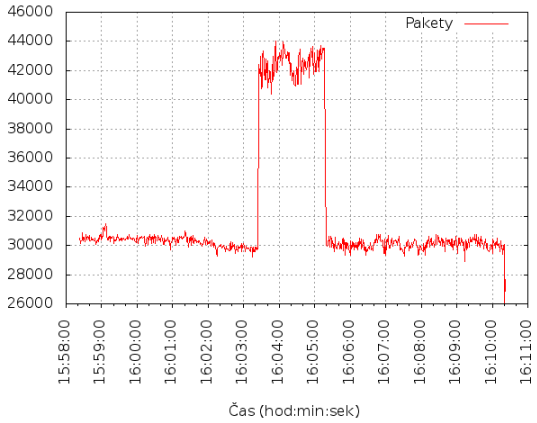


Obrázek C.2: Počet bytů.

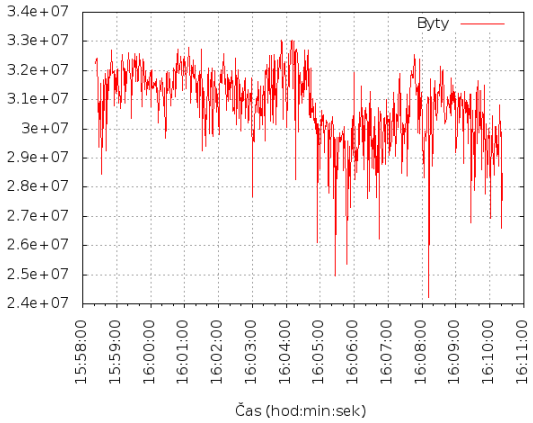


Obrázek C.3: Počet toků.

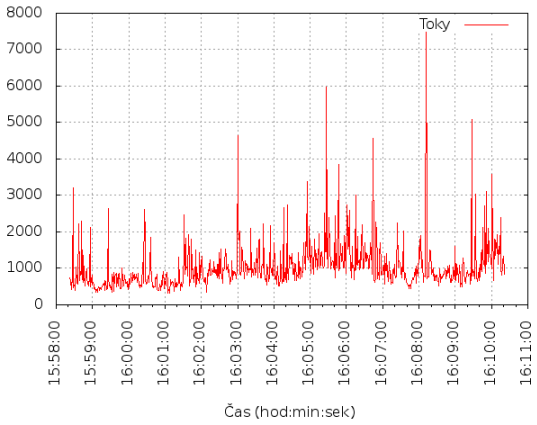
C.2 Útok ICMP flood



Obrázek C.4: Počet paketů.

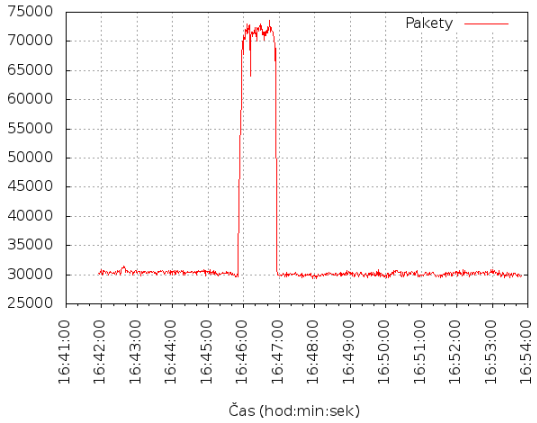


Obrázek C.5: Počet bytů.

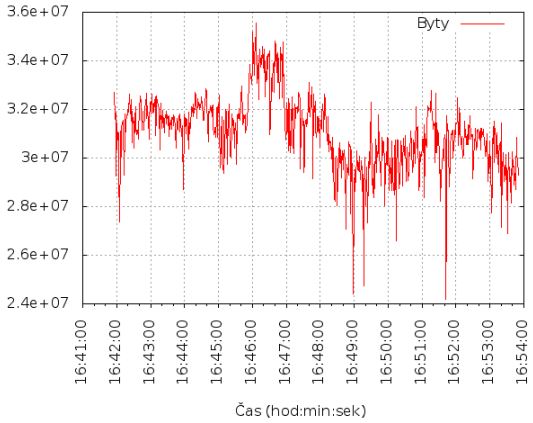


Obrázek C.6: Počet toků.

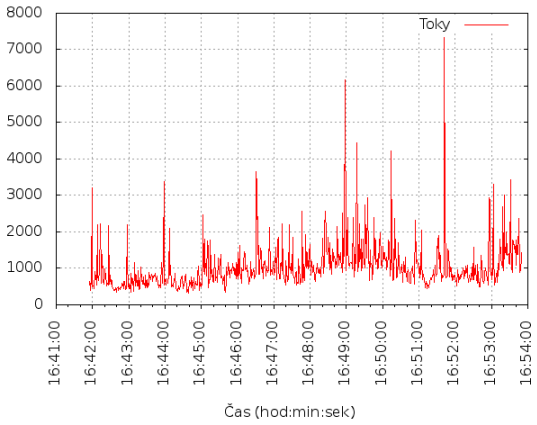
C.3 Útok HTTP flood



Obrázek C.7: Počet paketů.

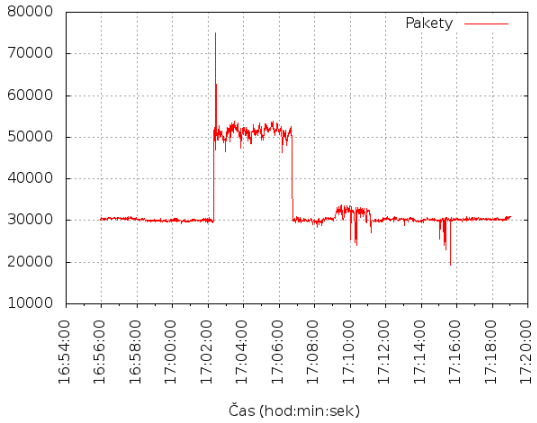


Obrázek C.8: Počet bytů.

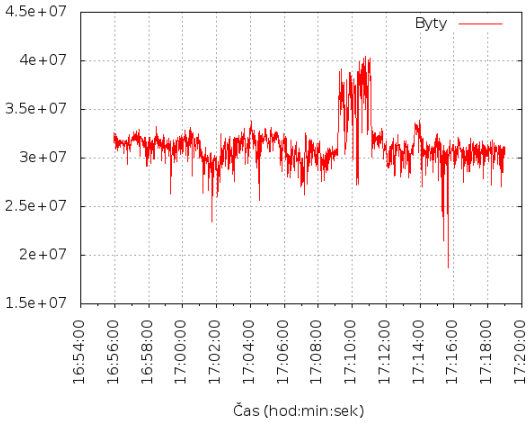


Obrázek C.9: Počet toků.

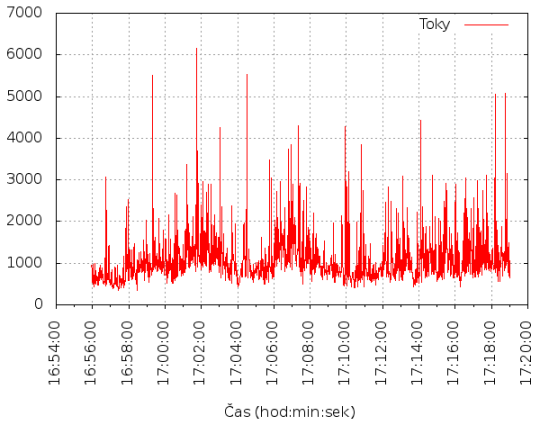
C.4 Útok UDP flood



Obrázek C.10: Počet paketů.

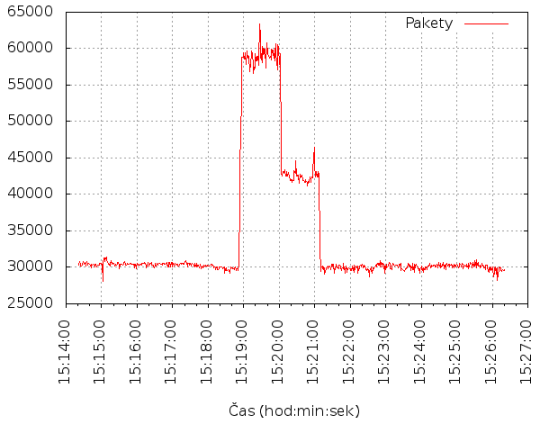


Obrázek C.11: Počet bytů.

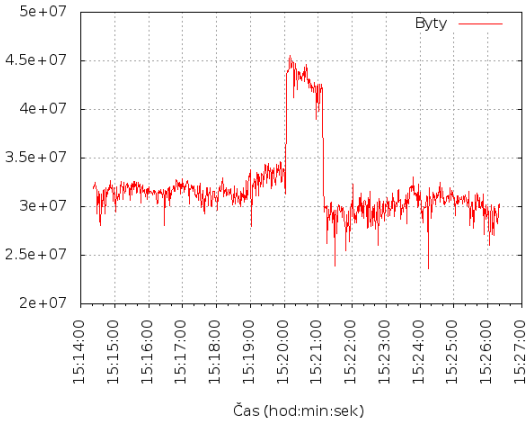


Obrázek C.12: Počet toků.

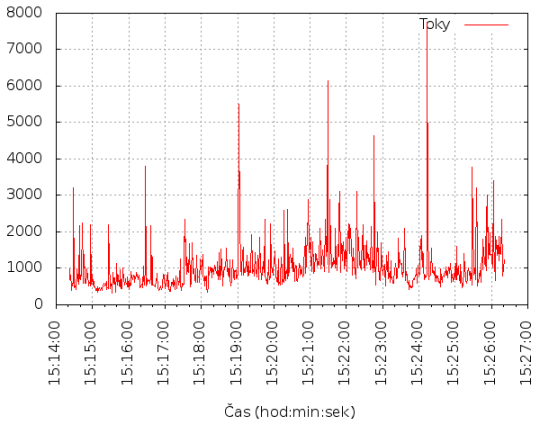
C.5 Útok TCP flood



Obrázek C.13: Počet paketů.



Obrázek C.14: Počet bytů.

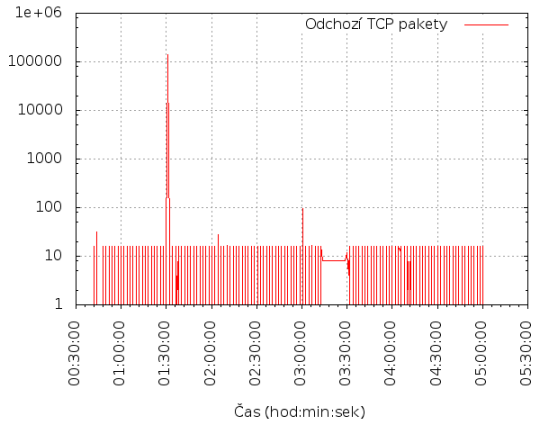


Obrázek C.15: Počet toků.

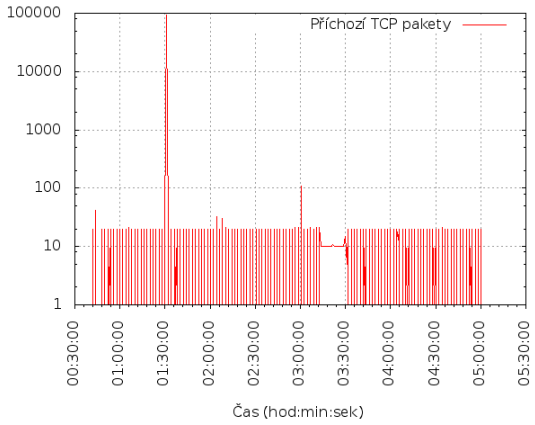
Příloha D

Grafy poměru odchozího a příchozího TCP provozu

Na grafech jde vidět, že odchozí a příchozí provoz se za normálních podmínek chová velmi podobně. Jde o pětihodinový úryvek z týdenního zaznamenaného provozu. Velký výkyv v počtu paketů je způsoben pravidelnou zálohou, která se takto projevovala každý den. Právě takové výkyvy by se metodou poměrů mezi odchozím a příchozím provozem odhalovaly velmi obtížně. Server však nebyl příliš vytížený a tak se zálohování projevilo více, než by tomu bylo na vytíženém serveru.



Obrázek D.1: Odchozí TCP provoz.



Obrázek D.2: Příchozí TCP provoz.